



CENTRO UNIVERSITÁRIO DE BRASÍLIA- UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

ROBERTO AUGUSTO FREITAS CAMPOS

AUTOMAÇÃO RESIDENCIAL UTILIZANDO ARDUINO E APLICAÇÃO WEB

BRASÍLIA – DF

1º SEMESTRE DE 2014

ROBERTO AUGUSTO FREITAS CAMPOS

AUTOMAÇÃO RESIDENCIAL UTILIZANDO ARDUINO E APLICAÇÃO WEB

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. MSc. Luciano Henrique Duque

Brasília

Junho, 2014

ROBERTO AUGUSTO FREITAS CAMPOS

AUTOMAÇÃO RESIDENCIAL UTILIZANDO ARDUINO E APLICAÇÃO WEB

Trabalho apresentado ao Centro
Universitário de Brasília
(UniCEUB) como pré-requisito
para a obtenção de Certificado de
Conclusão de Curso de Engenharia
de Computação.

Orientador: Prof. Luciano Henrique
Duque.

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -
FATECS.

Prof. Abiezer Amarília Fernandes
Coordenador do Curso

Banca Examinadora:

Prof. Luciano Henrique Duque
Mestre, UniCEUB

Prof^a. Ingrid Maria Dittert
Doutora, UniCEUB

Prof^a. Nathaly Sarasty Narváez
Mestra, UniCEUB

AGRADECIMENTOS

Agradeço, acima de tudo, a Deus, que me concedeu a graça de chegar ao fim de mais um desafio em minha vida;

Aos meus pais, José Augusto e Maria da Conceição, e às minhas irmãs, Fernanda e Lígia, pelo amor incondicional e pelo apoio prestado em momentos tristes e felizes, incentivando-me sempre ao crescimento pessoal e profissional;

À minha namorada e companheira de faculdade, Mayume, pelo companheirismo e pela compreensão de sempre e aos amigos de curso pela união prestada para alcançar o objetivo conjunto da aprovação.

Ao professor e orientador Prof. MSc. Luciano Henrique Duque, que com suas orientações me proporcionou a realização desse trabalho.

" *Uma Comunidade Inteligente é uma comunidade que fez um esforço consciente para usar a tecnologia da informação para transformar a vida e o trabalho dentro de seu território de forma significativa e fundamental, em vez de seguir uma forma incremental*"

California Institute for Smart Communities, 2001.

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO.....	13
1.1 - Motivação	14
1.2 - Objetivos	14
1.3 - Metodologia	15
1.4 - Resultados esperados.....	17
1.5 - Organização	17
 CAPÍTULO 2 REFERENCIALTEÓRICO.....	 19
2.1 - Histórico da Automação Residencial	19
2.2 - Domótica.....	21
2.3 - Automação Residencial: Realidade Hoje!.....	23
2.4 - Sistemas Integrados.....	23
2.5 - Dispositivos Utilizados.....	24
2.5.1 - A Plataforma Arduino	24
2.5.1.1 - Arduino UNO	25
2.5.1.2 - Arduino Mega	28
2.5.1.3 - IDE e linguagem de programação	31
2.5.2 - Servidor Web.....	32
2.5.2.1 - Apache.....	33
2.5.2.2 - Linguagem de Programação.....	34
2.5.3 – NFC.....	35
2.5.4 - Rádio Frequência (RF).....	37
2.5.5 – Relé.....	38
2.5.6 - Umidade do Solo	39
 CAPÍTULO 3 DESENVOLVIMENTO.....	 41
3.1 - Apresentação Geral do Projeto Proposto	41
3.2 - Descrição das etapas do sistemas integrados com controle centralizado	43
3.2.1 - Servidor Web.....	43
3.2.2 - Central de Controles e Comandos.....	44
3.2.2.1 - Interruptor Digital.....	45
3.2.2.2 - Controle de Iluminação por Dispositivos Móveis.....	49
3.2.2.3 - Acionamento do Portão Automático	51

3.3 - Descrição das Etapas do Sistema Autônomo	54
3.3.1 - Sistema de Irrigação	55
3.3.2 - Sistema de Filtragem da Piscina	57
3.3.3 - Sistema de Controle de Acesso.....	60
CAPÍTULO 4 Testes e resultados	63
4.1 - Erro 1	63
4.2 - Erro 2	66
4.3 - Erro 3	67
4.4 - Erro 4	67
CAPÍTULO 5 Conclusão.....	69
5.1 - Conclusão.....	69
5.2 - Sugestões para trabalhos futuros	70
REFERÊNCIAS	71
APÊNDICE A – Página inicial do servidor Web	73
APÊNDICE B – Código do controlador central	77
APÊNDICE C – Código do sistema autônomo de automação	79
APÊNDICE D – Código do leitor de NFC.....	82
APÊNDICE E – Código do interruptor digital	84

LISTA DE FIGURAS

FIGURA 2.1 – ARQUITETURA CENTRALIZADA DOS EQUIPAMENTOS	22
FIGURA 2.2 – ARDUINO UNO	25
FIGURA 2.3 – ESQUEMA ELÉTRICO DO ARDUINO UNO REVISÃO 3	27
FIGURA 2.4 – ARDUINO MEGA	28
FIGURA 2.5 – ESQUEMA ELÉTRICO DO ATMEGA 2560	31
FIGURA 2.6 – TRANSMISSOR E RECEPTOR RF - 433 MHZ	38
FIGURA 2.7 – DIAGRAMA DE UM RELÉ	39
FIGURA 2.8 – SENSOR DE UMIDADE DE SOLO	40
FIGURA 3.1 – SISTEMAS INTEGRADOS COM CONTROLE CENTRALIZADO	42
FIGURA 3.2 – SISTEMAS AUTÔNOMOS.....	42
FIGURA 3.3 – CONFIGURAÇÃO DE SERVIDOR VIRTUAL	44
FIGURA 3.4 – DISPLAY SENSÍVEL AO TOQUE	46
FIGURA 3.5 – TFT LCD MEGA SHIELD V2.0	47
FIGURA 3.6 – CODIFICAÇÃO DO ARDUINO MEGA (FONTE: AUTOR).....	48
FIGURA 3.7 – PÁGINA WEB DEMONSTRA A LUZ ACESA E APAGADA	49
FIGURA 3.8 – HTML DA PÁGINA WEB.....	50
FIGURA 3.9 – PARTE DO CÓDIGO DO CONTROLE DE ILUMINAÇÃO.....	51
FIGURA 3.10 – CENTRAL MICROPROCESSADA KX30	52
FIGURA 3.11 – BOTOEIRA EXTERNA.....	53
FIGURA 3.12 – PÁGINA WEB MOSTRANDO PORTÃO ABERTO E FECHADO	53
FIGURA 3.13 – CÓDIGO PARA ACIONAMENTO DO PORTÃO	54
FIGURA 3.14 – SISTEMA AUTÔNOMO DE AUTOMAÇÃO.....	55
FIGURA 3.15 – CÓDIGO GERADO PARA LER E CONVERTER A UMIDADE DO SOLO...	56
FIGURA 3.16 – PONTE H.....	57
FIGURA 3.17 – CÓDIGO QUE CONTROLA O ACIONAMENTO DA FILTRAGEM.....	58
FIGURA 3.18 – SHIELD RTC COM LCD	59
FIGURA 3.19 – SHIELD NFC	61
FIGURA 3.20 – PARTE DO CÓDIGO REFERENTE AO ACESSO DE USUÁRIOS	62
FIGURA 4.1 – LEITURAS CONSECUTIVAS DO NEXUS 4	64
FIGURA 4.2 – PARTE DO CÓDIGO DA BIBLIOTECA PN532_SPI.....	64
FIGURA 4.3 – LEITURAS CONSECUTIVAS DO NEXUS 4 – APÓS A CORREÇÃO	65
FIGURA 4.4 – ERRO DE SINCRONIZAÇÃO	66

LISTA DE TABELAS

TABELA 1 – RESUMO DO ARDUINO UNO	26
TABELA 2 – RESUMO DO ARDUINO MEGA 2560	28
TABELA 3 – QUOTA DE MERCADO DE TODOS OS SITES.....	34
TABELA 4 – MODELO DO FILTRO X CAPACIDADE POR HORA.....	58

LISTA DE ABREVIATURAS E NOMENCLATURA

V – Volts

NFC – *Near Field Communication* (comunicação por aproximação de campo)

IDE - *Integrated Development Environment* (ambiente de desenvolvimento integrado)

cm – centímetros

mA – miliampéres

AC – corrente alternada

PCI – placa de circuito impresso

E / S – entrada e saída

GPU - *Graphics Processing Unit* (unidade de processamento gráfico)

RAM - *Random Access Memory* (memória de acesso aleatório)

MB – megabite

LCD – *Liquid Crystal Display* (tela de cristal líquido)

HTML – *HyperText Markup Language* (linguagem de marcação de hipertexto)

PHP – *Php Hypertext Preprocessor* (preprocessador de hipertexto Php)

Web – rede de alcance mundial

RTC – *Real Time Clock* (relógio de tempo real)

RESUMO

O propósito do trabalho é utilizar um dispositivo móvel com acesso à internet, para automatizar serviços rotineiros nas residências. Uma aplicação web rodando num servidor disponibilizará o controle sobre a abertura e fechamento do portão automático, controle da iluminação da casa, controle de acesso aos ambientes por meio da tecnologia NFC, utilizada nos smartphones mais modernos; acionamento do sistema de irrigação do jardim e filtragem da água da piscina. Essa aplicação enviará dados a uma interface de hardware através de uma comunicação serial entre o computador e o ARDUINO UNO, onde esse interpretará os dados e controlará os serviços da residência. Para execução de algumas tarefas, basta o dispositivo móvel (tablets, smartphones ou notebooks) estar conectado à mesma rede local onde se encontra o servidor. Tanto através do wi-fi, como por cabo.

Palavras chaves: Automação, Arduino, Controle, NFC.

ABSTRACT

The purpose of the paper is to use mobile device with internet access, to automate routine services in homes. A web application running on a server provide control over the opening and closing of the automatic gate control home lighting, access control environments through NFC technology, used in most modern smartphones; Trigger garden irrigation and water filtration system of the pool. This application will send data to a hardware interface via a serial communication between the computer and the ARDUINO UNO, which interprets this data and monitor services from the residence. For some tasks, just the mobile device (tablets, smartphones or notebooks) is connected to the same LAN where the server. Either through wi-fi, and cable reception.

Keywords : Automation , Arduino , Control , NFC .

CAPÍTULO 1 INTRODUÇÃO

O hábito das pessoas está em constante transformação quando em relação às novas tecnologias e a alta exigência por conforto e segurança traz o tema da Automação Residencial para o seu lar.

A automação residencial, visto por muitos como algo do futuro, já é realidade em algumas residências hoje em dia, mas também algo muito caro, o que limita a sua generalização. Além disso, a convergência das tecnologias está levando ao conceito de ubiquidade, no qual dispositivos coordenam entre si para promover aos usuários acesso imediato aos serviços disponibilizados de forma transparente, visando aumentar a interação entre homem e sistemas automatizados. Pensando nisso, o foco desse projeto visa o segmento das aplicações dentro da automação residencial. Tendo em vista, minimizar custos com os equipamentos de controle, tornando mais comum e acessível as atividades rotineiras em nossas residências, no qual, coordenadas entre si, promove os usuários facilidade aos serviços disponibilizados, bem como, o simples ato de acender uma lâmpada, ligar uma bomba d'água, a irrigação de um jardim e, uma infinidade de atividades que possibilita aumentar a interação entre homem e sistemas automatizados.

Para tanto, é utilizado como base do projeto, o arduino, que é uma plataforma embarcada de código fonte livre. Em resumo, é uma placa simples com entradas e saídas (Banzi, 2010), e que oferece uma tecnologia de um custo muito baixo podendo ser usada na concepção de projetos baseados em microcontrolador (Monk, 2013), visando principalmente a economia financeira na hora de realizar um projeto de automação.

O arduino interage com o ambiente externo através de sensores e atuadores de diferentes tipos: os sensores captam informações importantes relacionadas àquele determinado lugar e informar ao arduino, que irá processá-las e acionar os atuadores para realizar as tarefas pré-determinadas ou requeridas por um usuário através de um dispositivo móvel que tenha acesso a uma rede local onde estará instalado o servidor.

Também podem ser acoplados aos arduinos os chamados *shield* (escudos) que são placas confeccionadas contendo outros dispositivos (por exemplo: displays LCD, receptores GPS, módulo de NFC) que se encaixam perfeitamente nos arduinos, obtendo funcionalidades suplementares (McRoberts, 2011).

Automação Residencial assim como outras inovações é primeiramente vista pelo usuário como algo moderno e de grande *status*. Com o conforto e a comodidade que ela proporciona, torna-se intrínseco (como a utilização dos *smartphones* no meio de nós) a sua utilização e um fator de geração de economia (Aureside, 2014).

1.1 Motivação

No cenário atual, as tecnologias estão provocando intensas alterações em todos os seguimentos da nossa vida. Elas vêm contribuindo, sem dúvida, para transformar o mundo. Atualmente, as pessoas levam uma vida muito corrida, tendo que realizar várias tarefas ao mesmo tempo, e nesse cenário, esquecer de fazer algo é muito comum, o que pode levar a gastos maiores financeiramente.

Automatizar pequenas tarefas, como por exemplo, colocar a água da piscina para filtrar sozinha, irrigação automática do jardim, ter um mecanismo que permite entrar em casa utilizando somente um smartphone sem a necessidade de um molho de chaves, que na correria do dia-a-dia esquecê-las é bem comum, traz um ganho de tempo para realizar outras tarefas, que para algumas pessoas são de suma importância. Além disso, desligar ou ligar uma luz, abrir ou fechar o portão, pelo smartphone, traz comodidade.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um ambiente residencial automatizado com o intuito de fornecer uma forma de comunicação entre sistemas, a

fim de trazer benefícios e comodidade ao usuário, possibilitando algumas funções de controle por dispositivos móveis como *smartphones*, *notebook* e *tablets*.

Os objetivos específicos deste projeto são compreendidos pelos seguintes fatores:

- Realizar pesquisas bibliográficas referentes ao arduino e sensores;
- Montar um display de LCD para comunicação sem fio com uma central de controles.
- Desenvolver uma central de controles para responder às requisições externas para o acionamento do portão automático ou/e para o controle de iluminação;
- Acoplar um RTC (relógio) para acionar a bomba da água da piscina;
- Montar sensores para avaliar a presença de água no solo;
- Desenvolver os *softwares* para rodar nos *hardwares* como projetado para determinada situação;
- Criar um ambiente WEB para o usuário interagir com a casa;
- Realizar testes para verificar se os sensores estão calibrados corretamente; Realizar testes de segurança no acesso a ambientes através da tecnologia NFC.

1.3 Metodologia

1. Realizar pesquisas bibliográficas sobre o arduino uno e o arduino MEGA, sensores de umidade do solo, display LCD, comunicador por radiofrequência, *shield* de NFC, módulo de relé e os serviços WEB;
2. Desenvolvimento dos *hardwares*:
 - Para o acionamento de luzes, utilizar um relé ligado à lâmpada e conectado ao arduino, que fará o acionamento deste;
 - Para o acionamento de fechadura elétrica será utilizado uma fonte de alimentação de 12V, que servirá para ativar a fechadura para abertura da porta, ligada a um outro relé

conectado ao arduino, que também estará conectado um módulo (*shield*) de NFC, para fazer as leituras de dispositivos que tenham essa tecnologia ou *tags* de NFC;

- Para a abertura de portão automático será utilizado o próprio mecanismo que se há nas placas do motor do portão, chamado botoeira, que funciona como uma chave de contato. Cada vez que é pressionada, ela abre o portão, se este estiver fechado, e fecha, se estiver aberto. E a chave nesse projeto vai ser um outro relay conectado ao arduino.
- Para o acionamento da bomba da água que visa filtrar a água da piscina, também será utilizado um relay conectado ao arduino, que todo dia terá um tempo de hora pré-determinado para ligar e desligar a bomba;
- Para monitorar a umidade do solo será utilizado um sensor de umidade, conectado ao arduino e quando verificado que o solo está seco, acionará um registro senoidal que libera a passagem de água para o sistema de irrigação.
- Para o interruptor digital com LCD, que visa ligar a luz do ambiente, será utilizado um arduino MEGA juntamente com um *shield* de LCD e um *Display* LCD de 3,2" e um componente de rádio frequência que tem a finalidade de comunicar com o outro arduino que controla o relay da lâmpada;

3. Desenvolvimento dos softwares:

- Foi utilizando a linguagem *Wiring* (Linguagem baseada em C/C++) para desenvolver os *softwares* que atuarão nos *hardwares*;

4. Criar um servidor WEB com o apache que terá a função de receber as requisições feitas pelos dispositivos móveis conectados a mesma rede do servidor, e desenvolver aplicações WEB utilizando a linguagem HTML, PHP e CSC, que permita ao usuário controlar de forma interativa os diversos recursos de automação;

5. Realizar testes operacionais dos hardwares e softwares a fim de se buscar uma melhor integração entre eles e uma confiabilidade tanto na segurança quanto na disponibilidade do sistema;
6. Criar um protótipo em versão miniaturizada de uma residência e automatizá-la para poder apresentar.

1.4 Resultados esperados

Espera-se com este projeto realizar um sistema de automação de uma residência e colocá-lo em funcionamento, permitindo o controle da iluminação, tanto por dispositivos móveis, quanto por um display de LCD que se comunica com a central de controle via ondas de radiofrequência; Permitir o acesso a ambientes por meio da tecnologia NFC; Controlar a abertura e fechamento do portão automático também por dispositivos móveis; Automatizar o sistema de filtragem da água da piscina; E ainda, irrigação do jardim averiguando-se a porcentagem de água no solo, atuando quando perceber necessidade.

1.5 Organização

A organização do projeto está dividida nos seguintes capítulos:

No capítulo 2 é apresentado o referencial teórico, com estudo sobre as tecnologias usadas na automação residencial;

O capítulo 3 apresenta o desenvolvimento do projeto, demonstrando suas funcionalidades e define os recursos utilizados para o desenvolvimento desse trabalho;

O capítulo 4 apresenta os testes realizados e os resultados obtidos;

O capítulo 5 sendo estes as considerações finais, contém as principais considerações obtidas ao longo do trabalho e são apresentadas as conclusões obtidas e sugestões para trabalhos futuros.

CAPÍTULO 2 REFERENCIALTEÓRICO

A robótica está intensamente relacionada com o ramo de automação e controle, quando falamos de engenharia moderna e cada vez vai crescendo mais o seu campo de estudos e de aplicação. O crescimento nas últimas décadas foi tão grande em pesquisas nessa área e também em automação das indústrias que não é exagero dizer que a engenharia eletrônica divide-se em três áreas: eletrônica analógica, eletrônica digital e em controle de sistemas, que inclui automação e robótica (Pazos, 2002).

2.1 Histórico da Automação Residencial

De acordo com Matias Bortoluzzi, o marco inicial da evolução tecnológica está pontado na década de 50, com a revolução industrial na Inglaterra, quando ocorreu a substituição da maneira de produção feudal pelo modelo mecanizado de produção capitalista. A descoberta da eletricidade também é outro marco histórico importante que deve ser levado em consideração juntamente com a popularização da sua utilização na 2ª revolução industrial no fim do século XIX. A partir desses momentos, Bortoluzzi cita as descobertas importantes que ajudaram na evolução tecnológica. São eles:

1876: Foi criada o telefone por Alexander Graham Bell através de uma comunicação entre um cômodo e outro;

1879: Thomas Edison criou a lâmpada;

1888: Heinrich Hertz foi precursor na transmissão de códigos utilizando ondas de rádio pelo ar;

1904: John Ambrose Fleming cria a válvula, permitindo o advento da eletrônica e a criação de equipamentos como a televisão;

1946: Foi construído o ENIAC, o primeiro computador digital eletrônico de grande escala, por John Mauchly e John Eckart Jr;

1947: A criação do transistor gerou o fim da era das válvulas, sendo descoberto por William Bradford Shockley, John Bardeen e Walter Houser Brattain. Os transistores são componentes semicondutores mais confiáveis e eficientes;

1956: A Ericsson cria o celular, chamado de Ericsson MTA (Mobile Telephony A). O aparelho móvel tinha que ser levado em um carro, porque pesava quase 40 quilos;

1977: Lançamento do primeiro computador com teclado integrado e capaz de gerar gráficos em cores;

1989: Início da popularização da internet e do celular.

A automação industrial se antepôs a automação residencial e graças a grande evolução que a microeletrônica teve com os Controladores Lógicos Programáveis (CLPs), oriundos da década de 60, várias empresas de tecnologia que trabalhavam na automação industrial mudaram seu foco para automação residencial sem perceber que essa última, tinha características diferentes da primeira. Enquanto na automação industrial eram criados equipamentos específicos para um controle seguro imune a falhas e com respostas imediatas aos comandos e alta perfeição. Já na automação residencial esses critérios não precisam ser levados com tanto preciosismo, porém, os dispositivos desenvolvidos necessitam de um acabamento mais fino e com interfaces que permitem que o usuário consiga o que deseja sem ter que recorrer a um manual de instruções, por exemplo.

Já o marco inicial da automação residencial pode ser considerado na década de 70 com o lançamento do protocolo X-10 nos Estados Unidos, que foram os primeiros módulos inteligentes que usavam a rede elétrica residencial como o meio de comunicação entre os vários equipamentos de automação. Essa tecnologia denominada *Power Line Carrier* (PLC) permitia automatizar uma residência sem ter que alterar a infraestrutura da rede elétrica da residência.

Com os computadores pessoais sendo cada vez mais populares na década de 80, pensou-se em utilizá-los como centrais de automação e controle. Porém, tinha uma grande desvantagem em utilizar para tal fim pois era necessário manter o PC constantemente ligado causando um consumo de energia muito grande. Uma outra desvantagem está relacionada a centralização do serviço, tudo rodando no mesmo PC, pode haver uma falha e comprometer todo o funcionamento do sistema de

automação. Com esses problemas, cria-se a ideia de se partir para o desenvolvimento de dispositivos que utilizam microcontrolador embarcado e micro processados para evitar a elevado consumo de energia, e abandonando-se a utilização dos computadores pessoais.

Concorrentemente foram incorporadas várias outras tecnologias à automação residencial como as comunicações sem fio, que possibilitam ao usuário o monitoramento e o controle de qualquer lugar da sua residência. (Bortoluzzi, 2013)

2.2 Domótica

O termo domótica é originado da junção das palavras Domus, que em latim significa casa, e robótica, que representa uma tecnologia capaz de controlar todos os ambientes de uma residência através de um só equipamento, incluindo temperatura, luminosidade, som, segurança, dentre outros, ou seja, automação residencial (Bolzani, 2004).

De acordo com Brugnara (2008), “a domótica é um recurso utilizado para controle de um ou mais aparelhos eletrônicos por meio de uma central computadorizada”.

Para a proposta descrita dá ênfase a uma melhor qualidade de vida para as pessoas, trazendo conforto e segurança juntamente com uma redução de gastos com os comandos centralizados de várias funções da residência como água, sistema de segurança, luz, dentre outros, é o principal objetivo do processo da domótica (Angel, 1993).

A domótica se utiliza de inúmeros equipamentos espalhados pela residência para satisfazer as requisições feitas pelos usuários. Podemos dividir esses equipamentos em 3 grupos principais (Takiuchi, Melo, & Tonidandel, 2004):

Atuadores: controlam os equipamentos da residência como luz, portão automático e irrigação;

Sensores: capturam informações do ambiente como, por exemplo, umidade e presença;

Controladores: controlam todos os sensores e os atuadores a fim de realizar as requisições feitas pelos moradores das residências aos equipamentos de automação residencial.

A Figura 2.1 expõe o esquema de como os 3 grupos principais atuam.

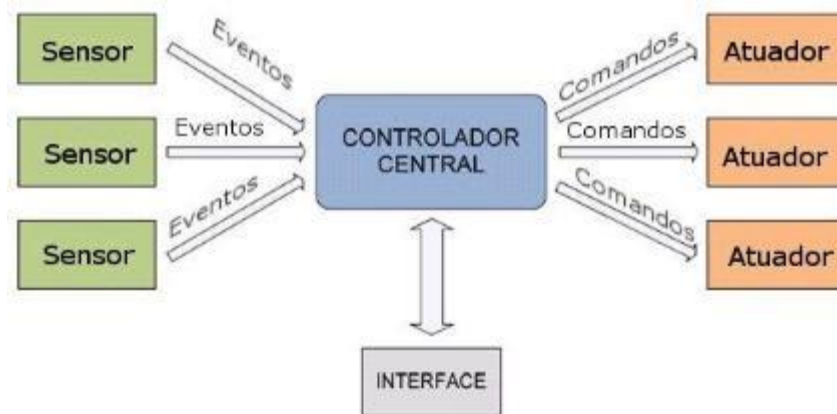


Figura 2.1 - Arquitetura centralizada dos equipamentos
(Fonte: Ferreira 2008)

Em se tratando de complexidade do sistema e níveis de integração, Angel (1993) divide os projetos de domótica em 3 tipos:

Sistemas autônomos: são sistemas isolados e independentes, isso é, em cada área da residência possui um equipamento de automação próprio, sendo controlado no próprio cômodo.

Sistemas integrados com controle centralizado: há somente um único controlador para todos os equipamentos de automação da residência, podendo controlar qualquer cômodo de qualquer lugar da casa.

Sistemas de automação complexos: para esse nível de automação, tem-se a necessidade da casa ser totalmente projetada para receber um alto grau de automação.

Com os custos dos equipamentos eletrônicos e de computadores caindo cada vez mais por causa do barateamento de vários componentes eletrônicos empregados na produção de hardwares, com as melhorias nas tecnologias usadas tanto na internet, como também na criação de softwares, era previsível que a automação industrial seria adaptada para as residências. Mas, com tantas diferenças

entre os ambientes residenciais e os industriais, foram necessários investimentos em pesquisas voltados para essa nova área (Bolzani, 2004).

Com a chegada dos microprocessadores e a criação de sensores e atuadores, a domótica teve um grande salto, pois esses equipamentos tornaram-se capazes de interagir com outros dispositivos necessitando de menos manutenção dos técnicos, ocasionando à automação residencial uma qualidade de controle muito grande e também a redução de espaços por eles utilizados (Bolzani, 2004).

2.3 Automação Residencial: Realidade Hoje!

Estamos sempre falando nas novas tecnologias que irão estar nas nossas casas futuramente. Mas, muito se fala no futuro e já está na hora de presenciarmos a automação residencial no presente. Vimos que alguns equipamentos eficientes são instalados para solucionar problemas pontuais como que aleatoriamente, o que acaba levando a frustrações para os clientes que pretendiam ter a automação residencial, pois acabam vivendo com um sistema autônomo, que não interage com outros equipamentos e em muitos casos tem uma operacionalidade não intuitiva (Aureside, 2014).

2.4 Sistemas Integrados

A integração entre dispositivos de uma automação residencial é a principal preocupação de projetistas e instaladores desses sistemas. Apesar dos mais modernos produtos oferecidos no mercado terem uma tecnologia complexa e avançada, disponibilizam uma interface interativa com os usuários para facilitar a sua operacionalidade.

Mas, quando vários dispositivos trabalham sem se comunicar entre eles geram uma desordem operacional na maioria das vezes, pois um sistema pode estar

fazendo a mesma coisa do outro ou até mesmo se conflitando na realização de suas funções, gerando assim um custo duplicado ao cliente e desnecessário.

A importância de se planejar um projeto de automação residencial desde a fase do projeto da construção da residência é de extrema importância, pois prevendo um cabeamento estruturado corretamente, o custo final desta integração de dispositivos fica bastante pequeno. Mas, quando não é possível a realização dessa etapa citada a cima, é necessário fazer algumas improvisações que eleva o custo do projeto e também o desperdício, resultando ao cliente um prejuízo financeiro e dificultando a operacionalidade do sistema. (Aureside, 2014)

2.5 Dispositivos Utilizados

Não tem como falar de automação residencial sem utilizar diversos dispositivos, sejam eles atuadores, sensores e controladores.

2.5.1 A Plataforma Arduino

O arduino foi originalmente projetado como um recurso para auxiliar os estudantes no ensino, mas em 2005 ele foi comercialmente lançado por Massimo Banzi e David Cuartielles, tornando-se um produto de sucesso entre fabricantes e estudantes devido a sua fácil utilização e a durabilidade que ele proporciona (Monk, 2013).

O arduino é um dispositivo formado principalmente por 2 componentes básicos: a placa arduino, que é o elemento de hardware utilizado para construir seus objetos e a IDE (Integrated Development Environment) do arduino, que é um programa executado no computador por onde escrevemos o código (chamado de *sketch*) e que fará o *upload* para a placa arduino. (Banzi, 2010).

O arduino é um pequeno computador capaz de processar informações de dispositivos e componentes externos conectados a ele através das suas entradas e

saídas de acordo com o que programamos. Também conhecemos o arduino como plataforma de computação física ou embarcada, isto é, um dispositivo de hardware com um sistema de software interagindo com o ambiente (McRoberts, 2011).

Existem diversas placas de arduino e todas utilizam a mesma linguagem para ser programada, mas a versão UNO é a mais utilizada disparadamente e a maioria usa as mesmas conexões com o ambiente externo, o que permite usar qualquer modelo facilmente (Monk, 2013).

Tanto o software, como o hardware do arduino são de fonte aberta a qualquer pessoa, isso quer dizer que tanto os códigos como os esquemas e também os projetos podem ser usados livremente por qualquer um e para qualquer propósito (McRoberts, 2011).

2.5.1.1 Arduino UNO

Existem muitas variantes do arduino diferentes e a versão mais recente delas é o arduino Uno, ilustrado na Figura 2.2.

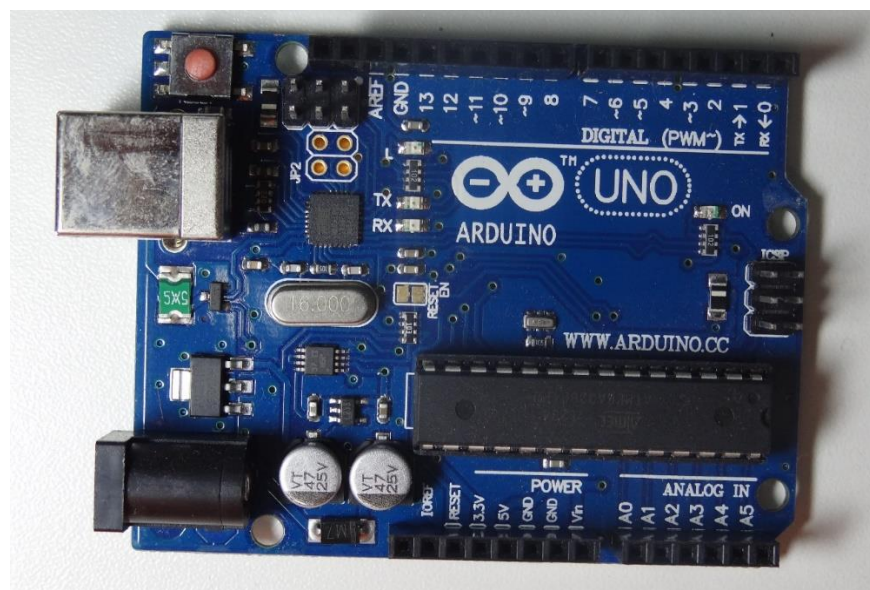


Figura 2.2 - Arduino Uno
(Fonte: autor)

A versão que popularizou essa marca foi a anterior, a Duemilanove (2009 em italiano) e provavelmente será a que mais terá projetos para arduino na internet. Mas, o mais popular e talvez o mais versátil seja o arduino Uno. O que torna esse sistema belo é que podemos criar qualquer coisa no arduino e depois transformar em algo permanente, precisando somente retirar o microcontrolador da placa e inseri-lo em uma placa de circuito própria (McRoberts, 2011).

O que difere em maior relevância o arduino Uno das versões anteriores é que o Uno utiliza um controlador USB diferente, alcançando uma comunicação mais rápida com o computador e a facilidade na instalação do drive (Monk, 2013). Na Tabela 1 apresenta as especificações do arduino Uno.

Tabela 1- Resumo do arduino Uno

Microcontrolador	ATmega328
Tensão de funcionamento	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Pinos digitais E / S	14 (dos quais 6 oferecem saída PWM)
Pinos de entrada analógica	6
Corrente DC por Pinos E / S	40 mA
Corrente DC para 3.3V	50 mA
Memória Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocidade do relógio	16 MHz

(Fonte: <http://arduino.cc/en/Main/ArduinoBoardUno>)

Do pino 0 ao pino 13 são considerados pinos digitais de entradas e saídas. Esses 14 pinos podem ser utilizados, conforme programado no *sketch* que for criado. Já do outro lado do arduino Uno, tem 6 pinos analógicos podendo ser utilizados somente como entradas, que são do 0 ao 5 e recebem valores de tensão de algum sensor ligado a ele e os transforma em um número entre 0 e 1023. Dos 14 pinos digitais, 6 podem ser utilizados como saídas analógicas que são os pinos: 3, 5, 6, 9, 10 e 11 (Banzi, 2010).

O Arduino Uno tem uma série de facilidades para se comunicar com um computador, outro Arduino ou outros microcontroladores. O ATmega328 fornece comunicação serial, que está disponível nos pinos digitais 0 (RX) e 1 (TX). O ATmega16U2 fornece comunicação serial através da USB e aparece como uma porta COM virtual para o software no computador. O firmware 16U2 usa os drivers COM USB padrão e nenhum driver externo é necessário. O software Arduino inclui um monitor serial que permite que dados simples de texto sejam enviados do computador para a placa Arduino. Os LEDs RX e TX da placa piscam quando os dados estão sendo transmitidos através do chip USB-to-serial e conexão USB para o computador (Arduino.cc, s.d.)

A Figura 2.3 apresenta o esquema elétrico do arduino uno revisão 3, em detalhes.

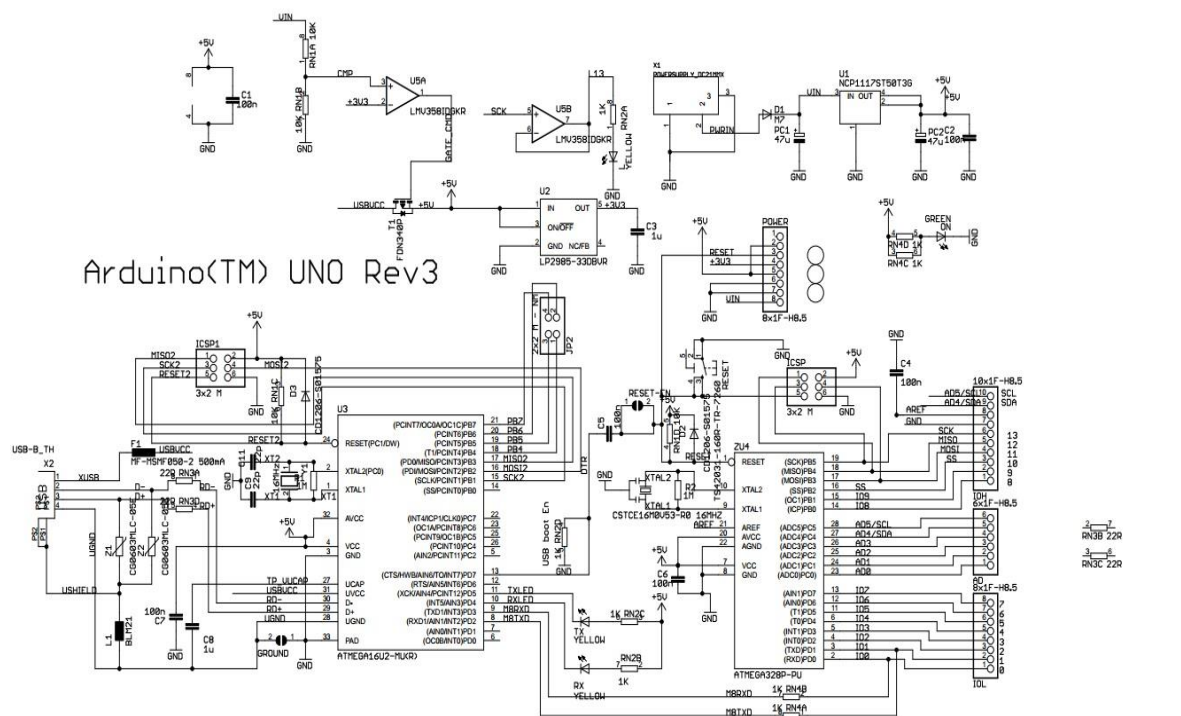


Figura 2.3 - Esquema elétrico do arduino Uno Revisão 3
(Fonte: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf)

2.5.1.2 Arduino Mega

O arduino Mega é o que tem a mais alta performance entre todos os modelos de arduino e também tem a maior quantidade de portas de entradas e saídas. A Figura 2.4 ilustra um arduino Mega e suas portas.

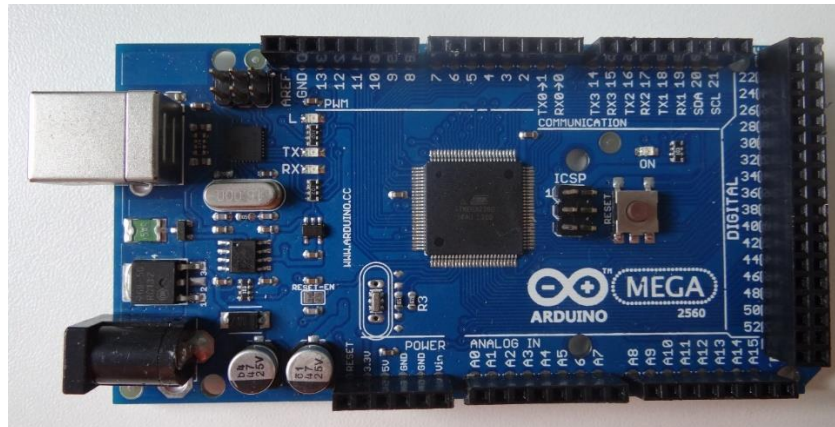


Figura 2.4 - Arduino Mega
(Fonte: Autor)

Um dos fatos que o torna um feito engenhoso na sua criação foi que todos os conectores laterais continuam compatíveis com os do arduino Uno, disponibilizando a ele todos os *shields* feitos para o Uno (Monk, 2013).

A Tabela 2 apresenta as características do arduino Mega.

Tabela 2- Resumo do arduino Mega 2560

Microcontrolador	ATmega2560
Tensão de funcionamento	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Pinos digitais E / S	54 (dos quais 15 oferecem saída PWM)
Pinos de entrada analógica	16
Corrente DC por Pinos E / S	40 mA
Corrente DC para 3.3V	50 mA
Memória Flash	256 KB
SRAM	8 KB
EEPROM	4 KB

Velocidade do relógio	16 MHz
-----------------------	--------

(Fonte: <http://arduino.cc/en/Main/ArduinoBoardMega2560>)

O Arduino Mega pode ser alimentado através da conexão USB ou com uma fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. A placa pode operar com um fornecimento externo de 6 a 20V. Se fornecido com menos de 7V, no entanto, o pino de 5V pode fornecer menos de cinco volts e a placa pode ser instável. Se usar mais do que 12V, o regulador de voltagem pode superaquecer e danificar a placa. O intervalo recomendado é de 7 a 12V (Arduino.cc, s.d.).

Os pinos de alimentação são os seguintes, de acordo com o site oficial do arduino:

VIN. A tensão de entrada para a placa Arduino quando se está usando uma fonte de alimentação externa (ao contrário de 5V a partir da conexão USB ou outra fonte de alimentação regulada). Você pode fornecer tensão por este pino ou, se o fornecimento de tensão através da tomada de energia, acessá-lo através deste pino.

5V. Este pino gera um tensão de 5V regulada pela placa. A placa pode ser alimentada com energia a partir da tomada de energia DC (7 - 12V), o conector USB (5V), ou o pino VIN da placa (7-12V).

3V3. Uma fonte de 3,3V gerado pelo regulador *on-board* onde a corrente máxima é de 50 mA.

GND. Pinos negativos.

O ATmega2560 tem 256 KB de memória *flash* para armazenamento de código (dos quais 8 KB é usado para o *bootloader*), 8 KB de SRAM e 4 KB de EEPROM (que pode ser lido e escrito com a EEPROM).

Cada um dos 54 pinos digitais do Mega pode ser utilizado como uma entrada ou uma saída, usando as funções `pinMode()`, `digitalWrite()`, e `digitalRead()`. Eles operam com 5V. Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor *pull-up* interno (desconectado por padrão) de 20-50 kOhms. Além disso, alguns pinos têm funções específicas:

Serial: 0 (RX) e 1 (TX); Série 1: 19 (RX) e 18 (TX); Série 2: 17 (RX) e 16 (TX); Série 3: 15 (RX) e 14 (TX). Usados para recepção (RX) e de transmissão de dados em série (TX) TTL. Pinos 0 e 1 também são ligados aos pinos correspondentes do ATmega16U2 USB-TTL *chip serial*.

Interrupções: Estes pinos podem ser configurados para ativar uma interrupção de um valor baixo, uma borda de subida ou queda, ou uma mudança de valor. Pinos 2, 3, 18, 19, 20 e 21.

PWM: Os pinos de 2 a 13 e de 44 a 46 fornecerem saída PWM de 8 bits.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) Estes pinos suportam comunicação SPI utilizando a biblioteca SPI. Os pinos SPI também são divididos no cabeçalho ICSP, que é fisicamente compatível com o Uno.

LED: Há um LED conectado ao pino digital 13. Quando o pino é de HIGH, o LED está ligado, quando o pino é LOW, ele está desligado.

O Mega2560 tem 16 entradas analógicas, cada uma das quais com 10 bits de resolução. Por padrão elas tem até 5V, embora seja possível mudar o limite superior de sua faixa (Arduino.cc, s.d.). Na Figura 2.5, temos o esquema elétrico simplificado do arduino Mega.

2.5.1.3 IDE e linguagem de programação

Fazer o upload (o carregamento) desse código (sketch) para o arduino através da conexão USB, aguardando alguns instantes a sua reinicialização;

E por fim, o arduino irá executar o *sketch* escrito;

O Arduino utiliza uma linguagem de programação simples, adaptada da linguagem *Processing*, que é chamada de *Wiring*. O que torna essa linguagem simples é o fato de quando pressionado o botão de *upload*, o código escrito é traduzido para a linguagem C (normalmente de difícil compreensão por quem está iniciando) (Banzi, 2010).

2.5.2 Servidor Web

Os servidores Web tem uma grande importância na troca de informações e como essas informações se apresentam em todo o planeta. Com o tempo, esses servidores vêm complementando as informações dos jornais, revista, rádios e outros meios de comunicação, e cada vez mais substituindo-os pela facilidade de acesso. Atualmente na Web existem mais de 28 milhões de servidores (Netcraft) acessíveis ao público geral, e este número está crescendo cada vez mais. Além desses servidores, existem também os que são destinados para intranets ou até mesmo para residências de uso privativo, podendo a conta passar pra milhares ou talvez milhões (Hoag, 2002).

Quando se acessa um site da Web pelos navegadores disponíveis no mercado é utilizado para a troca de informações entre o navegador e o servidor o protocolo chamado HTTP (HyperText Transfer Protocol, na tradução: Protocolo de Transmissão de Hipertexto). O objetivo do desenvolvimento desse protocolo era de mostrar o conteúdo do servidor Web nos navegadores dos usuários. Com o aumento crescente na utilização dos servidores Web e dos usuários requisitando informações, as empresas tiveram que mudar a maneira como as informações eram alteradas. Antigamente uma equipe responsável pela manutenção colocavam no site, realizava todas as alterações desejadas e reenviava o conteúdo novo ao servidor. Agora essas informações são alteradas de forma dinâmica a partir dos navegadores dos usuários, sem a necessidade da utilização da base de dados. Foi daí que projetaram o PHP (Personal Home Page, Página Pessoal), protocolo que manipula as informações/dados e os programas nos servidores Web, enviando em seguida os

dados ao navegador, permitindo a cada usuário que seja disponível as informações de forma personalizada. Existem também várias criações iguais a essas, permitindo páginas dinâmicas, como o Perl, C/C++, JavaScript e o ASP (Hoag, 2002).

2.5.2.1 Apache

O apache é um servidor Web que foi projetado por uma pesquisa da NCSA (National Center for Supercomputing Applications) da Universidade de Illinois na década de 1990. Rob McCool criou um programa baseado em plataforma UNIX de servidor Web simples para ser executado como um serviço, contendo poucas funções. Alguns anos depois, em 1994, um grupo de desenvolvedores resolveram se unir e organizar as extensões e correções no projeto do McCool, chamado agora pelo termo 'patch' (consertado) referindo-se ao antigo servidor Web, mas esse nome não durou muito e foi substituído pelo nome que temos hoje: APACHE. Em dezembro de 1995, esse grupo de desenvolvedores do apache lançaram a versão Apache 1.0 com a realização de teste de todas as extensões e correções, e no ano seguinte esse servidor se tornou o servidor Web mais conhecido e utilizado da internet (Hoag, 2002).

Em termos de sites ativos, o Apache continua em uma posição muito mais forte, com uma quota de 52% do mercado, em comparação com o Servidor Web da Microsoft 11%. Uma proporção significativamente mais elevada de sites em servidores Web Apache estão ativos: 26% de todos os sites do Apache foram consideradas ativas, enquanto que apenas 6% dos da Microsoft estavam ativos. O Apache está entre os sites mais movimentados, onde tem uma quota de 53% do mercado, enquanto nginx tem 18% e a Microsoft tem 12%. Embora apenas 3% usam o software de servidor web do Google, o domínio do Google está entre os sites mais movimentados com uma presença em 8 dos 10 principais sites (Netcraft, 2014).

A última versão do Apache (2.4.9) foi lançada em 17 de março de 2014. The Apache Software Foundation descreve isso como representando quinze anos de inovação pelo projeto. No entanto, ainda é comum para muitos sites utilizar a versão 2.2, ou mesmo versões mais antigas. Embora Apache 2.4.8 não foi lançado, a versão de desenvolvimento (Apache/2.4.8-dev) foi encontrada em 675 locais durante esta

pesquisa, que ocorreu em março. Quase todos esses sites estavam funcionando em servidores FreeBSD que pertenciam a vários projetos Apache, principalmente Apache HTTPD Apache e OpenOffice. A Tabela 3 mostra o comparativo entre os principais servidores Webs da internet (Netcraft, 2014).

Tabela 3- Quota de mercado de todos os sites

Desenvolvedor	Março 2014	Por cento	Abril 2014	Por cento	Mudança
Apache	354.956.660	38,60%	361.853.003	37,74%	-0.87
Microsoft	286.014.566	31,10%	316.843.695	33,04%	1.94
nginx	143.095.181	15,56%	146.204.067	15,25%	-0.31
Google	20.960.422	2,28%	20.983.310	2,19%	-0.09

(Fonte: <http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html#more-14567>)

2.5.2.2 Linguagem de Programação

A *home page*, também conhecida como página Web é compreendida por ‘tags’, que são comandos especiais, e por textos de uma linguagem de programação denominada HTML (HyperText Markup Language). A simplicidade dessa linguagem é baseada na finalidade de formatar o texto mostrado e criar relações entre outras *home pages*, com isso, cria-se documentos com o conceito de hipertextos. Quando um navegador mostra uma página Web, ele analisa o texto da página e busca por caracteres especiais que são chamados de tags. Essas tags é que irão dizer como a informação contida na página deverá ser mostrada na tela para o usuário. As tags são sinais do tipo “< >” e “< />” e podem informar várias formatações ao texto, como por exemplo: se o texto será exibido em qual cor ou se será em negrito, itálico ou sublinhado, se será um endereço de outra página, entre outras coisas (Ramalho, 1996).

A linguagem HTML foi originalmente criada no Laboratório Europeu de Física de Partículas (CERN) por Tim Bern Lee. Com a NCSA Mosaic a popularização da Web cresceu e com isso puxou a popularização da HTML também. Com o aparecimento de vários navegadores que usavam o HTML para navegar pela Web,

formaram grupos com o objetivo de uma padronização para o HTML (Argonavis, 2004).

Em 1994, foi criada uma linguagem de programação chamada PHP (Persona Home Page) por Rasmus Lerdorf, com o objetivo de desenvolver páginas dinâmicas utilizando um conjunto de *scripts*. Com o crescimento das funcionalidades dessa linguagem, o criador achou necessário fazer uma implementação C para permitir que outros desenvolvedores criassem aplicações para Web de forma muito mais simples. E em 1995, Rasmus disponibilizou seu código na internet com outras pessoas com o intuito de receber ajudas e correções de 'bugs'. A segunda versão do PHP foi lançada em 1997, contendo aproximadamente 1% de todos os domínios da internet utilizando o PHP. O PHP, versão 3, teve como princípio um projeto acadêmico voltado para a área de comércio eletrônico. Os estudantes Andi Gutmans e Seev Suraski resolveram ajudar Rasmus a aprimorar o PHP reescrevendo o código-fonte, lançado oficialmente em junho de 1998 (Dall'Oglio, 2007).

2.5.3 NFC

Near Field Communication (NFC) é uma tecnologia desenvolvida para a comunicação entre dispositivos, e sua comunicação sem fio, exige uma aproximação de campo de aproximadamente 4 cm para poder ser reconhecida, permitindo o compartilhamento de dados entre dispositivos Android e entre Android com tag NFC.

Existem alguns tipos de tags NFC, também chamadas de etiqueta, que variam em complexidade conforme as suas especificações. As tags mais simples só podem ser lidas e escritas diferente das mais complexas, que possuem um *hardware* criptografado e podem ser programáveis, podendo realizar operações matemáticas e até mesmo sendo usado na autenticação de pessoas a um determinado serviço ou acesso a um setor. As etiquetas desse último tipo permitem interações complexas com o código por ela executado, podendo ter uma variedade de formatos escritas nelas, porém, no Android as tags são baseadas no padrão NDEF (NFC Data Exchange Format) que é um padrão criado pelo NFC Fórum (Android, 2013).

Os principais modos de operação com que o Android suporta são os listados a seguir:

- **Modo de leitor / gravador:** nesse modo é permitido que as etiquetas e tags possam ser lidas e / ou escritas por dispositivos NFC;
- **Modo P2P (peer to peer):** Neste modo é permitido que os dispositivos NFC troquem dados entre si, utilizando uma a funcionalidade do Android Beam.
- **Modo de emulação de cartão:** Neste modo é permitindo que o dispositivo NFC simule uma etiqueta, permitindo que leitores de NFC externos acessem o dispositivo, como num terminal de ponto de vendas.

O Google Nexus 4 com Android, tem um chip NFC Broadcom dentro. A Google optou por utilizar BCM20793 chip controlador nfc da Broadcom no Nexus 4 feita pela LG. Até o momento, praticamente todos os celulares Android NFC utilizam chips NFC da NXP. Todos os BlackBerrys e o Nokia Lumia 610 tem um chip seguro de NFC. A Broadcom é o terceiro fornecedor de chips controladores de NFC para telefones móveis (Rostli, 2013).

Algumas vantagens da tecnologia (BROADCOM, 2012):

- Modo de detecção de baixa potência reduzindo o consumo de energia para maior duração da bateria;
- Sistema que permite ao chip obter energia do ambiente tanto para recarregar a bateria como para efetuar transações de comunicação;
- Integrado por componentes de menor tamanho em comparação com o mercado, torna o chip uma boa solução para menores custos de fabricação;
- Dentro de um único dispositivo integra as funções de transações baseadas em aplicações por identificação (ID), com o suporte simultâneo de transferência de dados com elementos de segurança;

- Conecta-se facilmente com o InConcert da Broadcom ® 4330, como Bluetooth e Wi-Fi para solução de conectividade abrangente;
- Cria opções de conexões mais rápida do mercado;
- Contém uma lista completa de software da Broadcom, incluindo middleware Maestro TM, que é uma solução para reduzir a complexidade de projetos.

Os chips da Broadcom não têm um elemento de segurança incorporado. No Google Nexus 4, o elemento de segurança é incorporado a partir do Broadcom ST33 da STMicroelectronics conectado ao chip controlador de NFC. O Broadcom em si não tem uma tecnologia de cartões inteligentes para a produção de elementos seguros. STMicroelectronics também fornece chips controladores de NFC (Rostli, 2013).

2.5.4 Rádiofrequência (RF)

A transferência de informações por rádiofrequência é um dos recursos tecnológico mais antigos das telecomunicações. Ela é capaz de proporcionar comunicações codificadas através sinais de ondas eletromagnéticas, que se propagam pelo espaço. Os principais elementos que permitem que dois pontos se comuniquem são:

- O Transmissor: Transforma sinais analógicos ou digitais em ondas eletromagnéticas, transmitindo-as para o espaço por meio de uma antena transmissora, que serão recebidas pelo receptor no outro ponto;
- O Meio de Transmissão: A onda de rádio frequência se propaga através do ar, podendo ultrapassar alguns obstáculos, porém, enfraquecendo seu sinal;
- O Receptor: tem a função de captar, por meio de uma antena, os sinais eletromagnéticos enviados pelo Transmissor e decodificá-los, convertendo em sinais digitais ou analógicos;

A Figura 2.6 ilustra os sensores de comunicação sem fio que utilizam a frequência 433.92 MHz para a comunicação de dados, tendo um alcance de até 150 metros (em condições perfeitas) a uma taxa de comunicação de 8.000 bps (WENSHING, 2013).

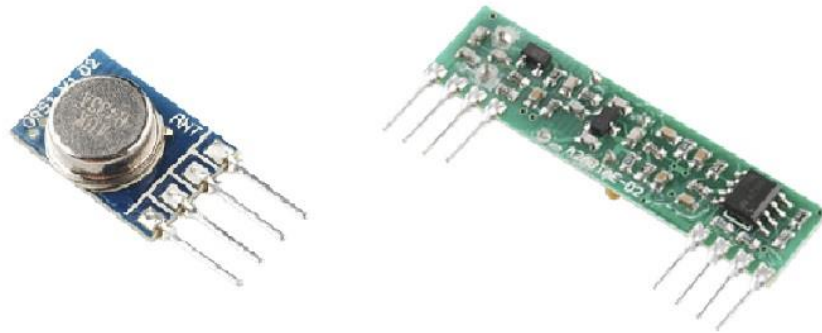


Figura 2.6 - Transmissor e Receptor RF - 433 MHz
(Fonte: Autor)

2.5.5 Relé

O Relé é um mecanismo eletromecânico surgiu em torno do século XIX, e é composto por um magneto móvel que se move, unido a 2 contatos feitos em metal. A maior utilização do relé foi no ramo de sistemas telefônicos em locais remotos quando existia as centrais telefônicas analógicas, e são considerados uma espécie de ascendente dos transistores. Os relés são considerados dispositivos muito lentos, pois o mesmo leva mais de um milésimo de segundo ser acionado e abrir ou fechar um circuito. Mas mesmo com esse ponto negativo, é bastante usado em grande maioria de sistemas eletrônicos que necessitam desse tipo de mecanismo de controle (Santos, 2013).

Os relés funcionam de maneira simples, considerando o seu mecanismo interior mostrado na Figura 2.7, eles funcionam através de corrente elétrica. Quando uma corrente passa pela bobina, ela gera um pequeno campo magnético capaz de atrair um material condutor fazendo com que o circuito fique no modo aberto ou fechando. Quando a corrente para de passar pela bobina, o campo magnético gerado por ela também cessa, e isso faz o relé retornar a sua posição original, estando aberto, ele fecha o circuito e vice versa. Os relés têm contatos chamados de NA

(normalmente aberto, quando não tem a passagem de corrente elétrica pela bobina), NF (normalmente fechados, quando existe a passagem de corrente elétrica pela bobina) e C (comum ou central, que faz a ligação entre o NA ou NF) (Santos, 2013).

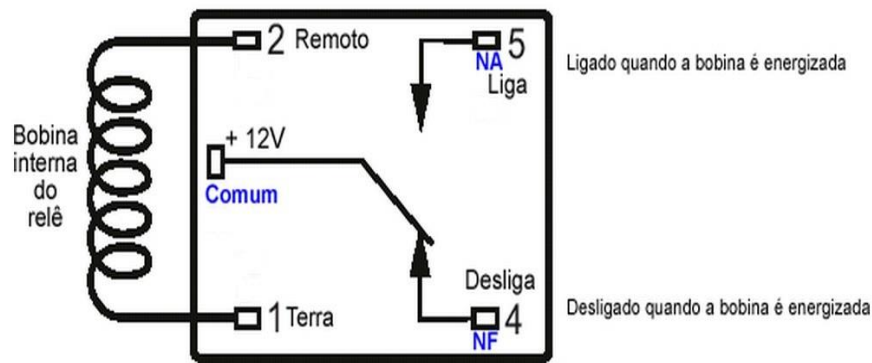


Figura 2.7 - Diagrama de um relé
(Adaptado de: <http://autosom.net/artigos/rele.htm>)

Os relés têm uma grande vantagem em relação aos seus componentes similares como o SCR (Retificador Controlado de Silício) e os TRIACS (Triode for Alternating Current), pois os relés estão totalmente isolados dos sistemas de controle, podendo trabalhar com valores de tensões diferentes. Mas, o desgaste precoce dos mecanismos interno do relé é a grande desvantagem de utilizá-los (Santos, 2013).

Os relés são compostos pelos seguintes materiais:

- Eletroímã (bobina) - constituído por fio de cobre em torno de um núcleo de ferro macio que fornece um caminho de baixa relutância para o fluxo magnético;
- Armadura de ferro móvel;
- Conjuntos de contatos;
- Mola de rearme;
- Terminais - estes podem variar dependendo da aplicação.

2.5.6 Umidade do Solo

Não somente a monitoração de níveis como pressão, vazão, temperatura e são variáveis fundamentais nos processos industriais, mas também outros indicadores são importantes nos processos industriais, e um deles é o nível de umidade. Esse monitoramento dos níveis de umidade pode, em grande parte das vezes, significar em uma qualidade maior do produto com um menor custo de fabricação ou até mesmo ter uma produtividade alavancada. Quando o excesso ou a escassez de umidade em qualquer situação influenciar ou até mesmo induzir física, química ou biologicamente nos processos, é de extrema necessidade que esses níveis de umidade sejam monitorados e controlados com precisão, para tanto deve-se utilizar um mecanismo de sensores de umidade (Thomazini & Albuquerque, 2005).

O sensor de umidade FC-28 mostrado na Figura 2.8 a seguir, faz leituras da quantidade de umidade presente no solo. Ele é ideal para criar dispositivos de medição de umidade em jardins ou vasos. Este sensor, utiliza as duas sondas para passar corrente através do solo, e faz a leitura da resistência para obter o nível de umidade. Quanto mais água, o solo se torna mais condutor de eletricidade (menos resistência), enquanto que o solo seco conduz menos eletricidade (maior resistência).

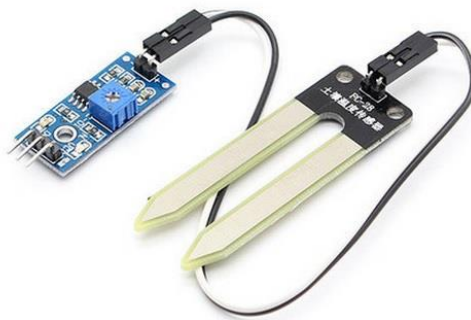


Figura 2.8 - Sensor de umidade de solo

(Fonte: <http://www.eletrogate.com/pd-89e60-modulo-sensor-de-umidade-de-solo-fc-28.html>)

Características:

Alimentação: 3.3 V ~ 5 V

Tensão de Sinal de saída: 0 V ~ 5 V

Interface: Analógica e Digital (analógica é mais precisa)

CI: LM393

CAPÍTULO 3 DESENVOLVIMENTO

Neste Capítulo será apresentado o desenvolvimento do protótipo do projeto proposto passo a passo tomando por referência os conceitos descritos no capítulo anterior.

3.1 Apresentação Geral do Projeto Proposto

O projeto está dividido basicamente em 2 dos 3 tipos que Angel citou no item 2.2, que são os sistemas integrados com controle centralizado e os sistemas autônomos.

A automação dos dispositivos do sistema integrado com controle centralizado são os relacionados abaixo:

- Controle de iluminação por acionamento através de dispositivos móveis e também por um interruptor digital, compreendido por um display sensível ao toque;
- Controle de acionamento do portão automático;

Para tanto, foi desenvolvido um servidor WEB somente local com o intuito de hospedar os sites que fará a interatividade do usuário com o sistema de automação. A Figuras 3.1 demonstra de forma simplificada como a interação entre o usuário e o objeto de automação ocorre. Onde o usuário realizará as requisições através do seu *smartphone* ao servidor Web, que esse por sua vez envia uma mensagem ao controlador central informando o desejo do usuário para que ele realize o acionamento do portão automático ou que ligue ou desligue a iluminação.

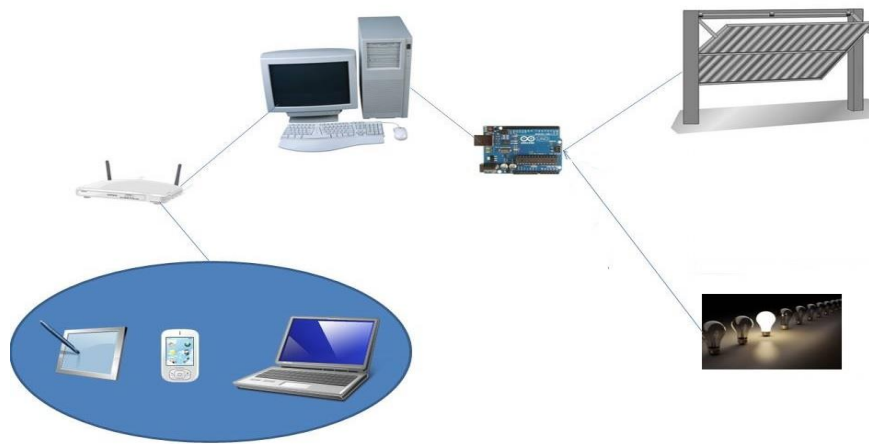


Figura 3.1 - sistemas integrados com controle centralizado
(Fonte: Autor)

Já os itens de automação residencial utilizando o conceito de sistemas autônomos são:

- Sistema de controle de acesso por meio de smartphone com tecnologia NFC;
- Controle automático e inteligente de irrigação de jardins;
- Acionamento automático de bomba d'água que faz a filtragem da água da piscina;

A próxima Figura 3.2 exemplifica os sistemas autônomos sem a necessidade de que o usuário envie comandos ao controlador desses sistemas.

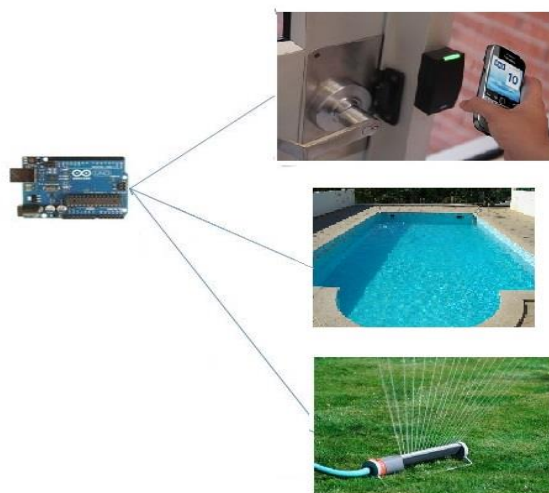


Figura 3.2 – Sistemas autônomos
(Fonte: autor)

3.2 Descrição das etapas do sistema integrado com controle centralizado

Como anteriormente citado, a integração entre os dispositivos é a grande preocupação de projetistas e instaladores desses sistemas, e é esse motivo que faz-se pensar em centralizá-los.

Estando o sistema de iluminação e o de controle do portão automático controlados em um único dispositivo, tornou-se conveniente a implementação de um ambiente interativo para que o usuário pudesse controlá-los com discricionariedade. Tendo como princípio que a tecnologia está ao alcance em quase todas as classes sociais, a utilização de dispositivos móveis que possibilitem o acesso a redes Wi-fi e que possam também navegar na internet, foi o que proporcionou a utilização de um servidor WEB residencial nesse projeto.

3.2.1 Servidor Web

A função do servidor WEB é a de proporcionar um ambiente de fácil interação entre o usuário e o que ele deseja realizar. Por se tratar de um serviço que não requer um alto poder de processamento, foi utilizado um notebook da Lenovo como Hardware do servidor Web, com as seguintes configurações:

Sistema operacional Microsoft Windows 8.1 de 64 bits;

Processador Intel Core I3-2328 com uma velocidade de 2.20GHz;

Memória RAM de 4,00 GB;

Disco rígido de 883 GB;

A escolha pelo APACHE (versão 2.0 de janeiro de 2004) se deu em conta de ser o mais popular servidor Web utilizado no mundo todo (de acordo com as pesquisas já citadas da Necraft), por ser um servidor gratuito e também por conter uma variedade muito grande de temas relacionados a ele na internet.

Faz-se de extrema necessidade a utilização de um mecanismo de comunicação entre o servidor Web e os dispositivos móveis (smartphones, tablets e computadores) com a função Wi-Fi, pois sem o mesmo não seria possível a realização do projeto proposto. Esse mecanismo utilizado é um roteador D-LINK modelo DSL-2730B que está conectado ao servidor Web através de uma conexão ethernet e se comunica com os outros dispositivos através da rede WI-FI.

Algumas configurações no roteador foram necessárias para encaminhar todas as requisições HTTP (porta 80) feitas por qualquer dispositivo conectados a ele ao servidor Web. A Figura 3.3 a seguir mostra a página de configuração do roteador para esse serviço e o endereço IP fixo do servidor Web:

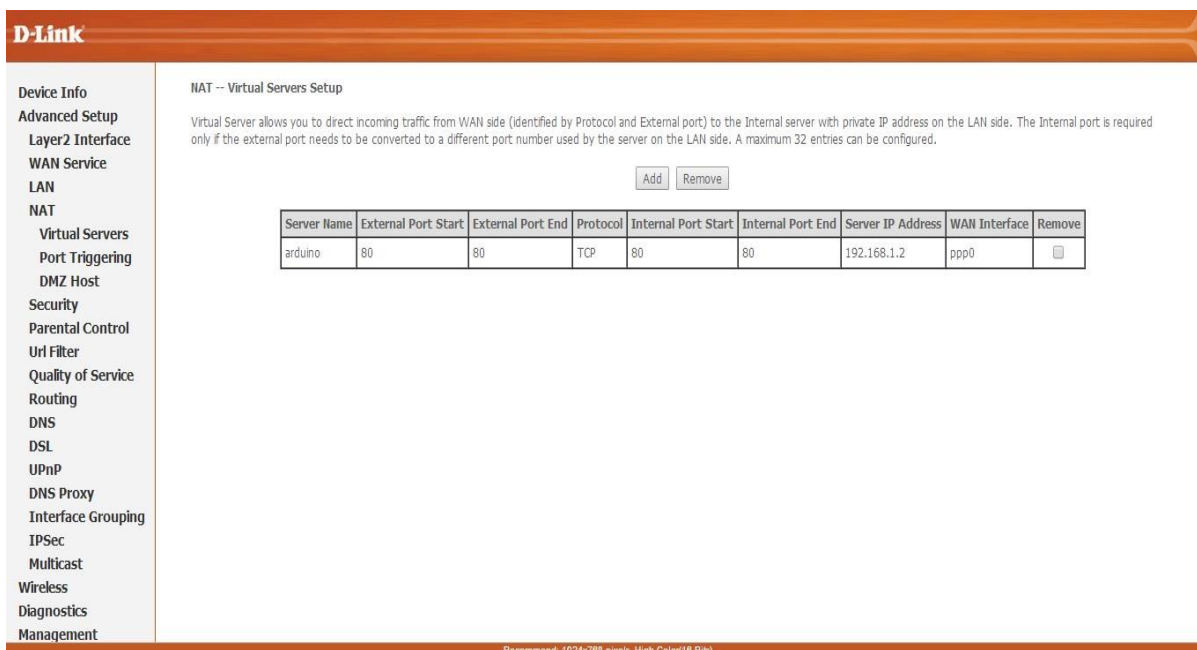


Figura 3.3 – Configuração de servidor virtual
(Fonte: autor)

3.2.2 Central de Controles e Comandos

Em se tratando de sistemas integrados de controle centralizado, a principal parte desse sistema de automação é sem dúvida o dispositivo que vai coordenar todas as ações, e no caso desse projeto, foi escolhido para esse papel o Arduino Uno.

O Arduino Uno, conforme comentado anteriormente, tem uma série de facilidades para se comunicar com computador, outro Arduino ou outros microcontroladores, e esse é o principal motivo pelo qual foi escolhido como o controlador central do projeto.

Por ele tem que passar todas as informações, e irá tratá-las e atuar, conforme solicitado. Essas informações podem chegar até ao controlador central de 2 formas: através da conexão USB com o computador e através da comunicação sem fio por rádiofrequência. Essa última, com a intenção somente de controlar a iluminação.

3.2.2.1 Interruptor Digital

O interruptor é um dispositivo simples, usado para abrir ou fechar circuitos elétricos e que basicamente envolvem o ligamento ou desligamento de energia elétrica.

Já o interruptor digital desenvolvido para realizar essa função nesse projeto é compreendido por um display LCD TFT de 3.2" com a tela resistiva, ilustrado na Figura 3.4, que detectam o toque graças à junção de duas placas sobrepostas. Quando a tela é tocada, tanto por uma caneta, quando por objetos sólidos ou até mesmo o dedo, é gerado um sinal elétrico, e conforme a posição do toque, uma placa encosta na outra.



Figura 3.4 – Display sensível ao toque
(Fonte: autor)

Essa tela tem uma resolução de 240 por 320 pixels e até 260 mil cores com interface de 16 bits paralela. Mas, para utilizar todas essas funções disponíveis nela, foi necessário um outro dispositivo para realizar o seu controle. É aí que entra o arduino Mega, referenciado anteriormente no capítulo 2.

A necessidade de se utilizar o controlador ATmega 2560 é basicamente que ele tem mais pinos de comunicação que o ATmega 328, sendo que para ligar o display é necessário ter 40 pinos. Entre o display e o arduino, foi utilizado um *shield* para facilitar a ligação entre os mesmos e que está ilustrado na Figura 3.5.

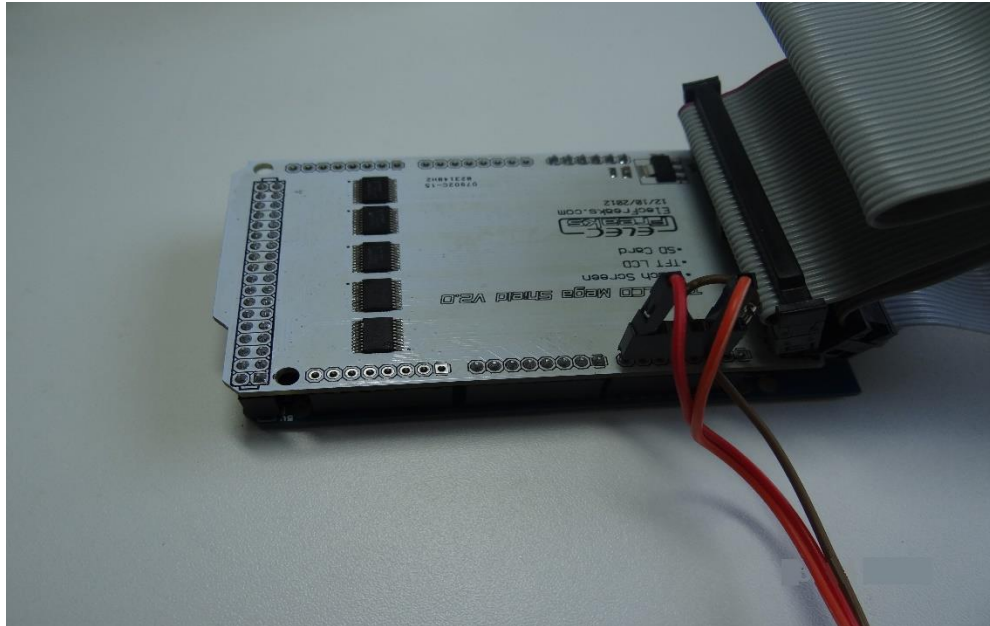


Figura 3.5 – TFT LCD Mega Shield V2.0
(Fonte: autor)

Outro dispositivo importante que engloba esse mecanismo de interruptor digital é o transmissor de rádiofrequência (mostrado na Figura 2.6), que também está conectado ao arduino Mega.

Com todos os dispositivos descritos acima integrados, o seu funcionamento ocorre da seguinte forma: No display é mostrado as opções “LIGAR” e “DESLIGAR” e a informação de qual opção o usuário apertou pela última vez, aparece no display conforme Figura 3.4. Após a seleção de uma dessas duas opções com um simples toque na tela, o arduino captará essa informação e irá codificá-la, enviando logo em seguida para o transmissor de rádiofrequência, que tem a função de transmitir pelo ar a informação codificada para a central de controles e comandos.

Para realização de todo esse sistema, foi necessário a utilização de algumas bibliotecas, as quais estão relacionadas a seguir:

- VirtualWire.h: utilizada para realizar o envio das informações pelo transmissor RF;
- UTFT.h: Essa biblioteca é responsável pelas imagens que serão apresentadas no display;
- UTouch.h: Para captar os toques nas telas, foram utilizados essa biblioteca e suas funções;

- UTFT_Buttons.h: Essa facilita a criação dos botões no display.

E parte do software gerado para controlar todo o sistema está demonstrado na Figura 3.6:

```
void loop()
{
    int but1, but2, botao_apertado;

    but1 = myButtons.addButton( 10, 20, 300, 70, "LIGAR");
    but2 = myButtons.addButton( 10, 100, 300, 70, "DESLIGAR");
    myButtons.drawButtons();
    myGLCD.print("VOCE APERTO:", 10, 200);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.setBackColor(VGA_WHITE);
    myGLCD.print("          ", 10, 220);

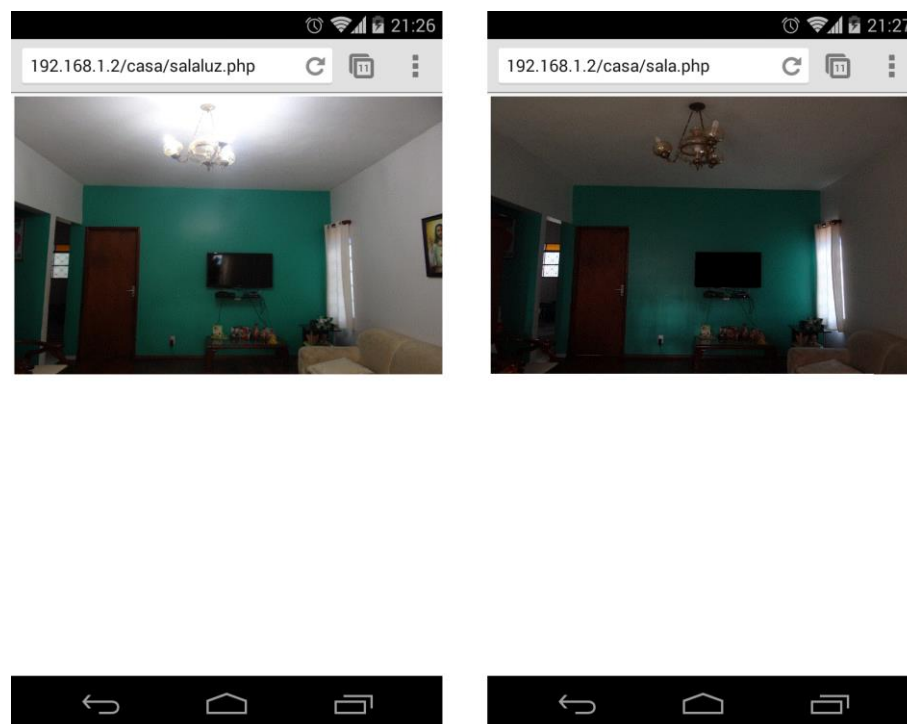
    while(1)
    {
        if (myTouch.dataAvailable() == true)
        {
            botao_apertado = myButtons.checkButtons();
            if (botao_apertado==but1)
            {
                myGLCD.print("LIGAR          ", 10, 220);
                const char *msg = "f";
                vw_send((uint8_t *)msg, strlen(msg));
                vw_wait_tx();
            }
            if (botao_apertado==but2)
            {
                myGLCD.print("DESLIGAR          ", 10, 220);
                const char *msg = "h";
                vw_send((uint8_t *)msg, strlen(msg));
                vw_wait_tx();
            }
        }
    }
}
```

Figura 3.6 – Codificação do arduino Mega (Fonte: autor)

3.2.2.2 Controle de Iluminação por Dispositivos Móveis

Uma outra maneira de comandar a ativação de uma lâmpada é por meio dos dispositivos móveis, que será feita por meio de uma conexão Wi-Fi entre o dispositivo e um roteador que por sua vez, estará conectado ao servidor Web.

Através da página Web (Figura 3.7) hospedada no servidor Web é disponibilizada a opção que comanda o acionamento ou não, da lâmpada com um simples toque no local indicado na foto, no caso uma luminária.



**Figura 3.7 – Página Web demonstra a luz acesa e apagada
(Fonte: autor)**

Para acessar tal página, basta digitar o endereço IP do servidor Web que esta hospedando-a, seguido de uma barra (/) e a palavra casa, ficando da seguinte forma: 192.168.1.2/casa, porém, esse endereço te levará a uma Home Page principal com 2 opções de seleção, uma delas leva ao controle de iluminação de uma sala. E a outra será explicada mais adiante.

O *back end* de todo esse sistema ocorre através de mensagens trocadas entre os dispositivos até chegar ao controlador central. Quando o usuário toca na área determinada da luminária, o navegador cria uma requisição para o servidor Web, que este, através de uma função disponibilizada pela linguagem PHP (citado no capítulo 2 deste trabalho), cria uma comunicação com o controlador central por meio da porta USB.

Na Figura 3.8 seguir é mostrado o código HTML utilizado para desenvolver a página Web e uma função em PHP, a qual tem uma importância fundamental para realizar a comunicação com o arduino, destacando que é ela que abre o canal de transferência e envia uma mensagem codificada, representada pela letra “f”. Quando o controlador central recebe um “f” pela comunicação serial, ele entende que é para desligar a luz, quando receber uma mensagem contendo a letra “h”, ele entende que é para acender a luz.

```

1  <!DOCTYPE html PUBLIC >
2  <head>
3    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4    <title>Untitled Document</title>
5  </head>
6
7  <body>
8    
9    <map name="Map" id="Map">
10     <area shape="rect" coords="373,2,680,216" href="salaluz.php" target="_top" />
11     <area shape="rect" coords="519,394,693,492" href="salaty.php" target="_top" />
12  <?php
13    $port = fopen('COM1', 'w+');
14    fwrite($port, "f");
15    fwrite($port, "e");
16    fclose($port);
17  ?>
18 </map>
19 </body>

```

Figura 3.8 – HTML da página Web
(Fonte: autor)

E para acender ou apagar a luz, o arduino utiliza o mecanismo de chaves visto no item 2.5.5, o relé, que está ligado na porta digital número 8. Conforme a mensagem recebida pela comunicação serial, o controlador central irá atuar ativando ou não o relé que está controlando a lâmpada indicada na página Web. Na Figura 3.9 a seguir é mostrado o código responsável pela execução do controle da iluminação.

```

    if(caracter == 'h')          // LIGA LUZ
    {
        digitalWrite(iluminacao, HIGH);
    }

//=====
    if(caracter == 'f')          // DESLIGA LUZ
    {
        digitalWrite(iluminacao, LOW);
    }

```

Figura 3.9 – Parte do código do controle de iluminação
(Fonte: autor)

3.2.2.3 Acionamento do Portão Automático

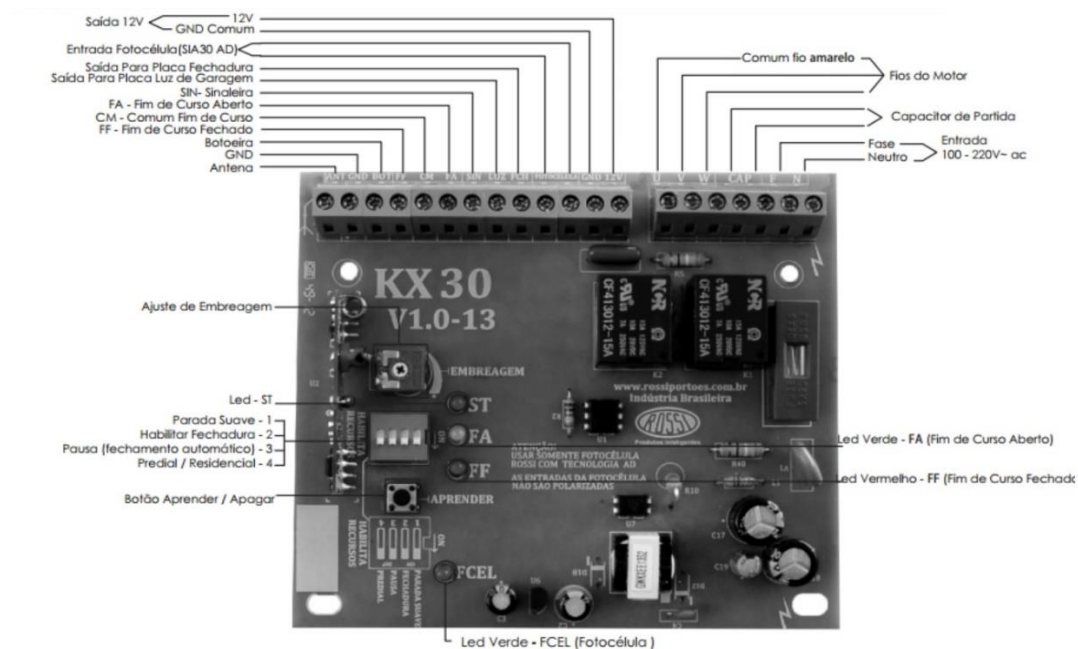
O último mecanismo que contempla a série é o acionamento do portão automático pelo mesmo meio que vimos no controle de iluminação.

A importância de se criar uma outra alternativa de ter esse controle, e não somente a do controle por rádiofrequência que a grande maioria dos portões automáticos fornecem, é a mesma que as novas tecnologias vem trazendo aos detentores de seus artifícios: a ubiquidade. De modo geral, parece que virou um hábito inquestionável o uso e a permanência constantemente pertos de seus *smarphones*, o que torna essa função de abrir ou fechar o portão muito mais ágil do que ter que ir ao carro pegar o controle do portão automático.

Na realização desse sistema, foi utilizado para realização de testes duas centrais microprocessadas, a KX30 V1.2-2013 (Figura 3.10) e a DZ-MX V1.0B, ambas fabricadas pela empresa Rossi, com características semelhantes a vários outros modelos disponíveis no mercado. No entanto, a principal diferença entre esses dois modelos citados é de que a DZ-MX é uma placa mais antiga, enquanto que a KX30 é uma versão mais moderna desta e por esse motivo, neste trabalho será utilizada a KX30 como estudo de pesquisa, que tem as seguintes características:

- Fonte automática 90 V – 240 V ~ ac;
- Fim de curso;

- Memória interna para 30 botões;
- Sistema de recepção HCS, anti-clonagem;
- Frequência de recepção 433 MHz;
- Função residencial e predial;
- Fechamento automático regulável (Pausa);
- Partida suave;
- Parada suave programável;
- Botoeira;
- Embreagem eletrônica regulável;
- Apaga controle individual;
- Entrada para sinaleira;
- Entrada para fotocélula AD;
- Saída para luz de garagem;
- Saída fechadura (FHC);



**Figura 3.10 – Central microprocessada KX30
(Fonte: Manual de instalação)**

E a característica mais importante dessas citadas anteriormente para esse projeto é a de ter a função Botoeira, detalhada na Figura 3.11. É ela a responsável por permitir que o objetivo descrito no capítulo 1 se realize.

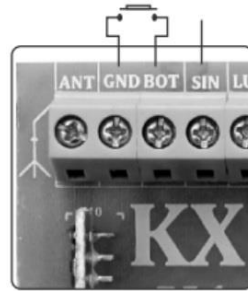


Figura 3.11 – Botoeira externa
(Fonte: manual de instalação)

A botoeira consiste em se instalar um botão de pulso (tipo campainha no borne (BOT) da central ou interligar uma Receptora. “A botoeira é utilizada para acionamento manual em guaritas, acionamento pelo interfone ou eventual necessidade de acionamento à distância por botão externo. ”

Tendo como referência essa informação contida no manual de instalação, e o conhecimento descrito sobre os relés no capítulo 2, foi possível trocar esse botão mecânico por um relé, que tem seu acionamento, eletronicamente.

Entretanto até chegar ao relé a requisição do usuário passa por alguns passos que serão descritos a seguir, começando pela página Web que é acessada por ele para poder controlar o acionamento do portão.

As imagens que podem ser vistas no dispositivo móvel são mostradas na Figura 3.12.

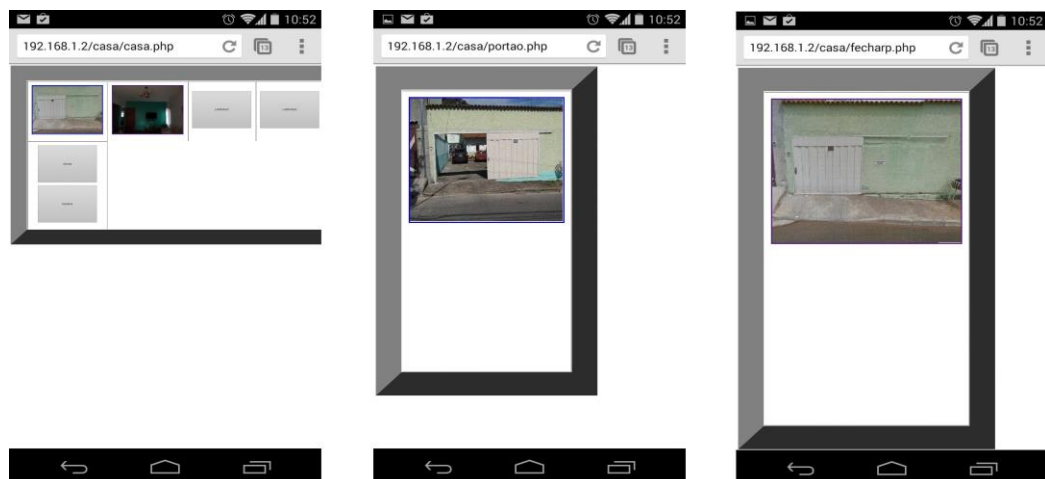


Figura 3.12 – Página Web mostrando portão aberto e fechado
(Fonte: autor)

Quando o usuário seleciona a opção do portão na página inicial, ele é encaminhado para a próxima página, a que mostra o portão aberto e nesse processo ocorre uma comunicação entre o servidor Web e o controlador central, igualmente ocorrido no controle de iluminação, mudando somente o conteúdo da mensagem enviada e consequentemente, abrindo o portão. A Figura 3.13 mostra uma parte do código para o acionamento do portão.

```
caracter = Serial.read();  
if(caracter == 'r')  
{  
    digitalWrite(portao, HIGH);  
    delay(1000);  
    digitalWrite(portao, LOW);  
}
```

Figura 3.13 – Código para acionamento do portão
(Fonte: autor)

O conteúdo da mensagem enviada ao arduino é o caractere “r”, que é interpretada como sendo a codificação necessária para o acionamento do relé pela porta digital número 12. E essa ativação temporária (1 segundo) faz fechar o circuito da botoeira e acionar o portão automático.

3.3 Descrição das Etapas do Sistema Autônomo

Por existir nas residências tarefas rotineiras que são sempre efetuadas da mesma maneira, é que foi pensado em alocar recursos dedicados para tais tarefas sem que o usuário precise interagir com o sistema. Tarefas como colocar todo dia a água da piscina para filtrar é de extrema importância para manter a boa qualidade dessa água boa para banho e ter um sistema de irrigação de jardim que analisa a quantidade de água predominante no solo vendo a necessidade de irrigar ou não, economizando água na residência do usuário desse sistema.

Mas, os dois motivos principais da realização desse sistema autônomo de automação são: o primeiro é o ganho de tempo que o morador vai ter em não realizar essas tarefas e o outro é que o usuário pode esquecer de realizar uma dessas tarefas e prejudicar a qualidade da água ou o aspecto de beleza do jardim.

A seguir será detalhado cada sistema separadamente, lembrando que os dois fazem parte de um único sistema, mostrado na Figura 3.14 a seguir.



Figura 3.14 Sistema autônomo de automação
(Fonte: autor)

3.3.1 Sistema de Irrigação

Esse sistema tem consistência exatamente no esquemático (Figura 1) de como os 3 grupos principais atuam: os sensores captam eventos, mandam para o controlador analisar e envia comandos para o atuador. Agora teremos uma análise de cada parte dessas, detalhadamente.

O sensor utilizado para captar sinais do ambiente é um sensor de umidade modelo FC-28, mostrado no canto esquerdo superior da Figura 3.14. Ele tem um mecanismo simples para captar a quantidade de água que existe no solo, utilizando a corrente elétrica em um dos seus terminais e no outro terminal ele irá captar o quanto dessa corrente consegue absorver. E isso ocorre pelo fato de quando o solo está seco, sem umidade alguma, não há passagem de corrente de um terminal ao outro, pois o

solo não consegue conduzir corrente nesse estado. Quanto mais o solo estiver úmido, maior será a sua condutividade e por consequência passará mais corrente elétrica de um terminal ao outro.

Esse dispositivo de captura atua como um sensor analógico, portanto é conectado à porta analógica 0 do arduino. Como já visto no referencial teórico deste trabalho, o arduino trabalha com tensões de 0 a 5 V, podendo ler valores das portas analógicas de 0 a 1023 por referência a esses valores de tensões.

No código escrito para o arduino, Figura 3.15, foi utilizado uma função capaz de gerar uma escala de 0 a 100 tendo como referência os valores lidos de 0 a 1023.

```
umidade_val = analogRead(sensorUmidade);  
saida_valor = map(umidade_val, 0, 1023, 100, 0);
```

Figura 3.15 Código gerado para ler e converter a umidade do solo
(Fonte: autor)

E foi convertido em porcentagem de água que o solo contém, tendo como objeto de análise que abaixo de 30%, o solo está seco, entre 31% e 70 % o solo está úmido, e acima de 71% o solo está muito úmido. O que faz com que o arduino mande o comando para o atuador liberar a água para irrigação abaixo desse valor de 30%.

O atuador em questão é uma válvula solenoide (dispositivo mostrado na Figura 3.14 no canto esquerdo inferior) que funciona com uma bobina que gera um campo eletromagnético, quando a corrente passa em um sentido, ela abre o fluxo para que a água passe, quando é invertido o sentido da corrente, ela interrompe esse fluxo. Para realizar essa tarefa de ficar invertendo os fluxos da corrente, foi necessário criar um sistema de transistores conhecido como ponte H, onde são utilizados 4 transistores ligados de forma que se consiga controlar esse fluxo por comandos eletrônicos emitidos pelo arduino. A seguir temos o esquema da ponte H (Figura 3.16) utilizada nesse projeto, onde os caminhos vermelho e verde são os caminhos que atuam na válvula solenoide, permitindo a irrigação do jardim ou cessando o fluxo de água. Para que a corrente faça o caminho vermelho, os transistores Q1 e Q4, que

estão ligados respectivamente nas portas digitais 6 e 9 do arduino, têm que estarem em seus estados elevados, e as digitais 7 e 8, ligadas nos respectivos transistores Q3 e Q2, têm que estarem nos seus estados baixos. A mesma lógica ocorre para que a corrente percorra o caminho de cor verde, fazendo que a válvula se aberta estiver, feche, e vice e versa. Porém, existe dois caminhos que devem ser evitados. São os mostrados nas cores azul e laranja, pois caso esse fenômeno ocorra, colocando as saídas digitais 6 e 7 do arduino em altas, os transistores Q1 e Q3 permitiria a passagem de corrente, havendo assim um curto no sistema e um superaquecimento podendo estragar os componentes utilizados e até mesmo o arduino.

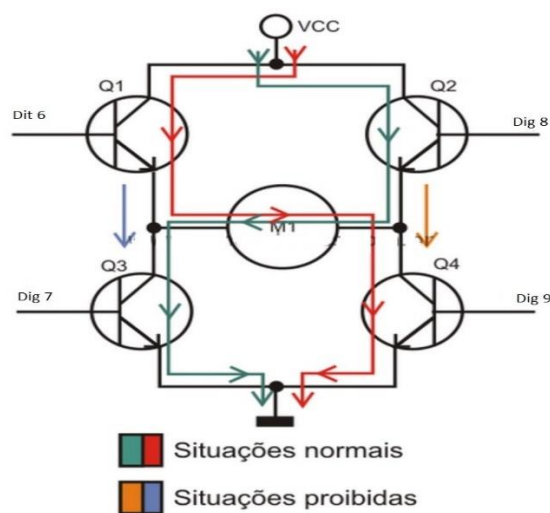


Figura 3.16 – Ponte H
(Fonte: autor)

3.3.2 Sistema de Filtragem da Piscina

Quem tem piscina em casa sabe o quão difícil é manter a água dela sempre limpa. E para tanto, existem vários procedimentos indispensáveis para o esse controle.

Para facilitar um desses procedimentos é que foi desenvolvido um sistema que liga o filtro automaticamente todos os dias por um intervalo de tempo pré-determinado, que varia de acordo com a quantidade de água que cabe na piscina e também com a capacidade que o filtro tem de filtrar a água por hora.

Nesse projeto foi utilizado um filtro fabricado pela Bestway, modelo 58148, que tem a capacidade de filtrar 2.006 litros por hora, conforme mostra a Tabela 4.

1.135 LIT (330GAL)	2.006 LIT (530 GAL)	3.028 LIT (800GAL)	4.921 LIT (1.300GAL) 5.678 LIT (1.500GAL)	7.949 LIT (2.100GAL) 9.463 LIT (2.500GAL)
58145	58148	58117	58122	58221
58145GB	58148GB	58117GB	58122GB	58221GB
58145GS	58149	58118	58123	58229
	58148GS	58117GS	58122GS	

Tabela 4 - Modelo do filtro X Capacidade por hora
(Fonte: Manual do produto)

Sabendo a capacidade que o filtro tem de filtragem por hora, basta dividir a quantidade de litros que a piscina suporta pela capacidade do filtro, que no caso do projeto proposto, a capacidade da piscina é de 2000 litros dividido pela capacidade do filtro, que é de 2006 litros por hora, necessitando estar ligado por aproximadamente uma hora.

```
void loop()
{
    DateTime agora = RTC.now();
    String relógio_data = "";
    String relógio_hora = "";

    int dia = agora.day();
    int mes = agora.month();
    int ano = agora.year();

    int hora = agora.hour();
    int minuto = agora.minute();
    int segundo = agora.second();

    if(hora >= 7 && hora <= 8)
    {
        digitalWrite(piscina,HIGH);
    }
    else
    {
        digitalWrite(piscina,LOW);
    }
}
```

Figura 3.17 - Código que controla o acionamento da filtragem
(Fonte: autor)

Na realização desse sistema, foi usada a biblioteca <RTCLib.h> com a função de manter sempre atualizado o horário, e o dispositivo RTC (Real Time Clock – Relógio de Tempo Real) que armazena a hora e nunca deixa desatualizar, mesmo com a falta de energia. Foi criada uma *shield* (Figura 3.18) pelo autor para integrar o RTC e o LCD no controlador central, com isso facilitar o manuseio e deixar mais bonito, sem uma grande quantidade de fio interligando-os.

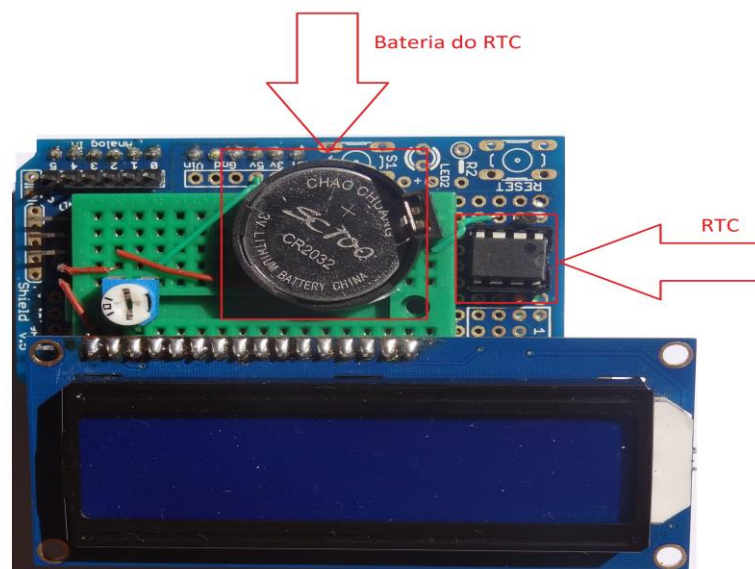


Figura 3.18 Shield RTC com LCD
(Fonte: autor)

No display de LCD, sempre ficará informando a data e a hora atual, pois é dessa informação que o controlador irá acionar a filtragem, e quando estiver acionado o filtro, um LED indicará o seu funcionamento, como pode ser visto na Figura 3.14.

O acionamento da bomba do filtro será realizado através de um relé ligado na saída digital 13 do arduino, que quando der a hora exata, irá colocá-la em nível lógico alto, fazendo com que o relé ative e ligue a bomba. O similar ocorre quando der a hora exata de desligar a bomba do filtro, colocando a porta digital 13 em nível lógico baixo.

3.3.3 Sistema de Controle de Acesso

Conforme o objetivo desse projeto é ter num único dispositivo o controle da casa e pensando também em que carregar chaves de casa consigo é algo incomodativo, que pode levar ao esquecimento delas em algum lugar e ficar trancado do lado de fora de sua própria residência é que foi proposto o desenvolvimento desse sistema, que consiste em o morador carrega consigo somente algo que ele provavelmente não irá esquecer, pois nos dias atuais, com a tecnologia invadindo e fazendo o usuário refém dela, as pessoas tomaram para si os smartphones como sendo um objeto de extrema importância comparadas até com as roupas que usamos no dia-a-dia.

Partindo desse princípio aonde em sua casa foi desenvolvido um sistema de acesso à residência por meio de uma tecnologia que alguns smartphones oferecem e já citada no item 2.5.3 do capítulo 2, que é a NFC. Foi utilizado como objeto de estudo nessa situação, o modelo de smartphone Nexus 4, com suas características referenciadas no capítulo 2 desse mesmo trabalho. Porém, podem existir moradores que não possuem tal tecnologia em seus smartphones. Então não será somente esse meio que o morador poderá acessar a sua residência, portando um cartão NFC com a ID cadastrada no sistema, esse também poderá acessá-la.

Como o sistema não conta com um banco de baterias ou algo similar que possa suprir a falta de energia quando ela acontecer foi utilizado uma fechadura que permite o acesso também por chaves.

O controlador utilizado para realizar as leituras das informações que provém do smartphone é um arduino Uno, porém, só com ele não seria possível tal feito e é por isso que a utilização de um *shield* de NFC modelo SHD-NFC (Figura 3.19) criado pela *Elecfreaks* faz-se necessário.

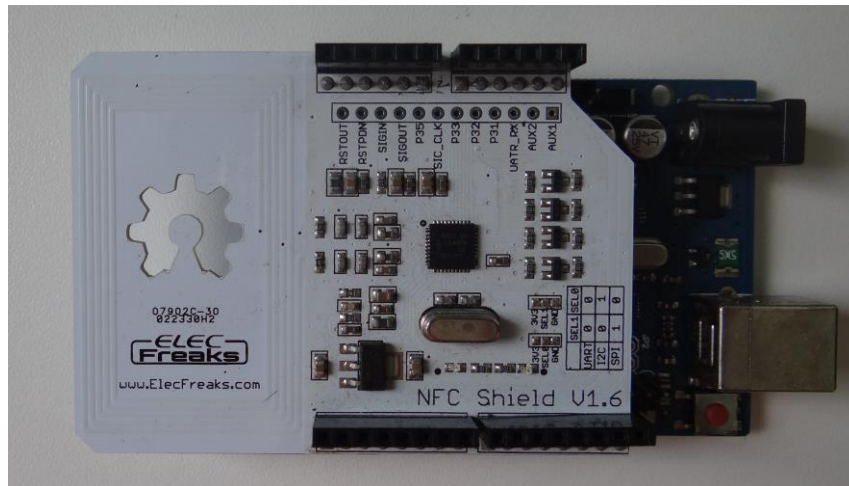


Figura 3.19 Shield NFC
(Fonte: autor)

Esse *shield* NFC operar em banda ISM de 13,56 MHz de frequência tendo um alcance de comunicação de até 10 cm podendo ser limitado pelo desenho da antena na PCI e a radiação de energia. O escudo NFC fornece todos os circuitos necessários para trabalhar com a biblioteca PN532, que fornece as funções necessárias para realizar as leituras e tem as seguintes características:

- Escudo compatível com Arduino, sem a necessidade de soldas;
- Interface SPI. Permitindo que vários outros pinos fiquem disponíveis para outras aplicações;
- Antena Construído em PCI;
- Suporta tanto 3.3V e 5V de operação;
- Soquete para conectar outros escudos;
- Cristal 27.12MHz;
- Pinos de E / S de PN532 de fácil acesso.

Com a função somente de ler os dados NFC, ele capta essas informações do smartphone ou de uma tag de NFC e verifica se ela tem permissão para entrar na residência.

```

void loop(void) {
  uint32_t id;
  id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);

  if(id == 2021683202 || id == 11184810 )
  {
    Serial.print("ID encontrada: "); Serial.println(id);
    digitalWrite(porta8, HIGH);
    delay(1000);
    digitalWrite(porta8, LOW);
  }
}

```

**Figura 3.20 Parte do código referente ao acesso de usuários
(Fonte: autor)**

Caso a identificação do *smartphone* lido ou do cartão seja as cadastradas no sistema, um LED verde acenderá indicando que o acesso foi permitido e a fechadura da porta destravará permitindo o acesso do morador à residência. Caso a identificação seja diferente das cadastradas, um LED na cor vermelha se acenderá indicando que o acesso não é permitido.

Quando à permissão de acesso, o arduino coloca sua porta digital 8 em nível alto fazendo por um período de um segundo que o relé conectado a essa porta se ative, fechando um circuito externo que permite uma passagem de corrente para a fechadura e assim fazendo-a abrir.

CAPÍTULO 4 Testes e resultados

Este capítulo abordará os temas referentes aos testes realizados para solução de problemas e erros que ocorreram no decorrer da construção do projeto. Ao todo foram relatados 4 erros de grande relevância nesse trabalho, que serão descritos a seguir.

4.1 Erro 1

O primeiro erro diz respeito ao sistema de controle de acesso por NFC utilizando o smartphone.

Quando a versão do sistema operacional (SO) instalado no smartphone era a versão 4.2 do Android (Jelly Bean), tudo funcionava normalmente. O sistema identificava a ID do smartphone e verificava se era aceita ou não. Porém, com a atualização desse SO para a versão 4.4 do Android (Kit Kat), a ID enviada pelo smartphone era alterada a cada leitura, não permitindo que pudesse cadastrá-lo como único.

A Figura 4.1 demonstra as leituras consecutivas realizadas pelo leitor de NFC após a atualização do sistema operacional.

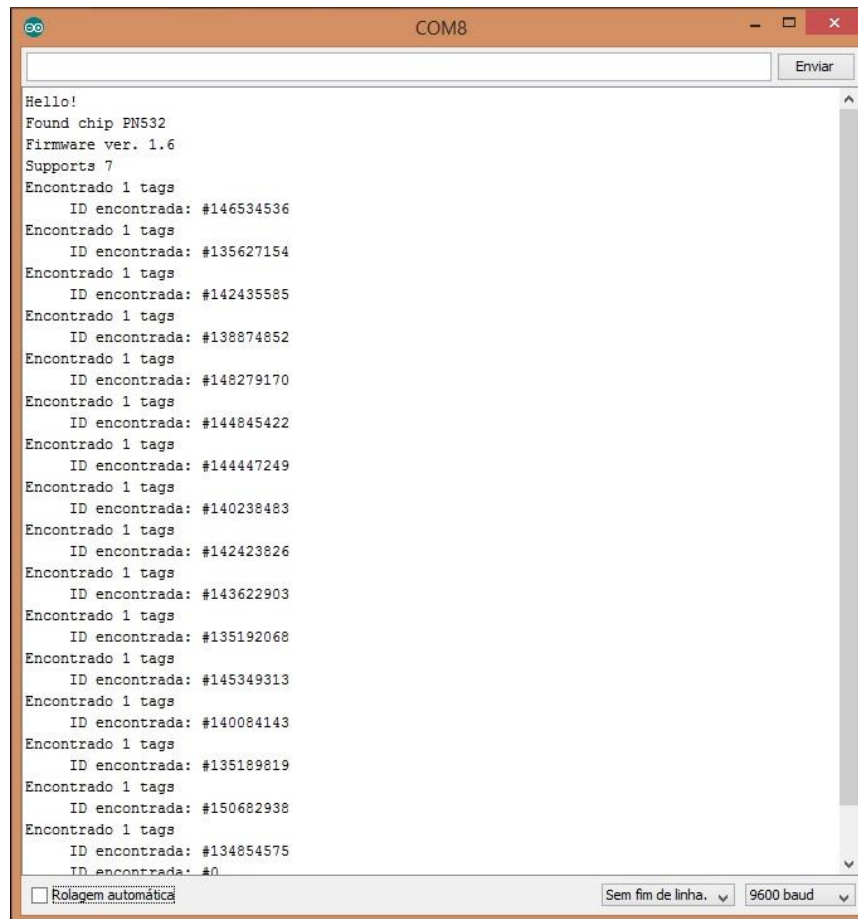


Figura 4.1 - Leituras consecutivas do Nexus 4 – após a atualização do SO
(Fonte: autor)

Esse erro ocorreu por causa de uma falha que continha no chip controlador de NFC do smartphone, comentado no item 2.5.3 deste trabalho e com a atualização do sistema, a Google tentou solucionar isso através de software, o que fez provocar esse erro. Isso ocorria, porque uma função da biblioteca PN532_SPI lia somente uma faixa de blocos de endereço e foi nessa faixa de blocos (Figura 4.2) que foi alterada com a nova atualização.

```

271  for (uint8_t i=0; i< 9; i++) {
272
273      Serial.print(" "); Serial.print(pn532_packetbuffer[11+i], HEX);
274  }
```

Figura 4.2 - Parte do código da biblioteca PN532_SPI
(Fonte: autor)

4.2 Erro 2

O erro 2 é referente ao sistema de iluminação, que não foi possível corrigi-lo, mas foi possível amenizá-lo através de software.

O erro é o seguinte: quando se liga a luz pelo interruptor digital, essa informação não é repassada ao servidor Web para que ele atualize a imagem que aparece na tela do smartphone, ficando assim, sem uma sincronização entre o que foi selecionado no interruptor digital, isto é, a real situação da lâmpada (acesa ou apagada) e a imagem que é mostrada quando acessamos a página Web para controlar a iluminação. A Figura 4.4 ilustra o erro em questão. Percebe-se que no interruptor digital está mostrando que a última seleção foi a de “desligar” e que realmente desligou a luz, porém, na imagem mostrada no smartphone, a página Web continua mostrando que a luz está acesa. Caso nessa situação seja apertado no *smartphone* para desligar a luz, nada acontecerá além da página mostrar uma imagem com a luz apagada.



Figura 4.4 Erro de sincronização
(Fonte: autor)

4.3 Erro 3

O erro 3 é referente a um sistema que não foi possível incorporar ao projeto, pois ele conflitava com um já existente no mesmo.

A princípio, um sistema de infravermelho para controlar a televisão estava previsto no projeto, porém, este estava conflitando com o sistema de rádio frequência.

O conflito ocorria não na transmissão dos sinais em simultâneo, mas era um problema de capacidade de hardware, pois o sistema de rádio frequência tem que ficar a todo momento checando se mensagens estão chegando e o sistema de infravermelho teria que parar essa checagem para poder enviar o sinal para o infravermelho. E por causa desse erro, houve a necessidade de optar entre um dos dois sistemas e o escolhido para fazer parte do projeto foi o da comunicação por rádio frequência.

4.4 Erro 4

Outro erro que ocorreu e que gerou uma outra solução foi a de utilizar um notebook para servir de servidor Web. A princípio foi utilizado um computador de baixo custo, Raspberry Pi B, porém, as configurações desse minicomputador não foram suficientes para atender com presteza as requisições que eram feitas pelos dispositivos móveis.

O Raspberry Pi versão b (versão testada) tem como as principais características as seguintes (Pi, 2014):

- Processador com velocidade de 700MHz (desempenho similar ao de um Pentium 2 300 MHz);
- 4 Processadores Gráficos integrados Unit (GPU);
- 512 MB de RAM;

Porém, elas não são suficientes para se ter uma resposta rápida às requisições, necessitando de se ter um processador mais veloz e com maior quantidade de memória RAM.

CAPÍTULO 5 Conclusão

Este capítulo abordará as conclusões referentes a realização de automação residencial utilizando arduino, juntamente com sugestões de implementação de novas funcionalidades e melhorias para trabalhos futuros.

5.1 Conclusão

Mesmo sendo uma área pouco explorada, a automação residencial oferece várias oportunidades para desenvolvimento e inovação, tanto na área de computação quanto na área de elétrica e eletrônica. A plataforma Arduino pode ser uma boa escolha para se desenvolver uma solução em automação residencial. A solução desenvolvida neste trabalho consegue atender aos requisitos básicos do controle residencial, integrando o sistema baseado no Arduino com um smartphone Android.

Nesse trabalho foi criado um sistema para automatizar alguns serviços residenciais e que foi alcançado o objetivo inicial proposto, tendo um bom funcionamento de todo o conjunto.

A utilização do Arduino como controlador dessas funções foi extremamente importante para o sucesso do controle e automação, pois percebeu-se que ele mantém uma boa eficácia em todos os testes realizados, além de ter uma confiabilidade muito grande. Porém, para algumas funções extras, como citado no item 4.3 por exemplo, mostrou-se um pouco limitado quando se acrescentavam mais funções à ele, exigindo a utilização de outros arduinos.

Uma grande característica desse sistema é a escalabilidade, podendo acrescentar novos ambientes ao controlador central, como pode ser visto tanto no código do arduino APÊNDICE B, como no código da página Web, APÊNDICE A.

A principal dificuldade encontrada no desenvolvimento deste trabalho foi o fato de trabalhar com tecnologias até então novas, como é o caso do NFC, o que

necessitou de um tempo maior de pesquisas e aprendizagem para desenvolver o sistema. Em relação à pesquisa acadêmica, essa tecnologia apresenta poucas referências bibliográficas, dificultando o recolhimento de dados para a revisão de literatura. Em contrapartida, as outras tecnologias utilizadas oferecem um grande acervo de material como exemplo que podem ser utilizados como base para o desenvolvimento das aplicações. Com base nos resultados apresentados conclui-se que o sistema que foi desenvolvido consegue atender as necessidades de controle em uma residência, eliminando o uso de diversos controles e centralizando algumas operações em um único dispositivo móvel, no caso um smartphone portando o a tecnologia NFC.

5.2 Sugestões para trabalhos futuros

Como sugestão para trabalhos futuros, caso deem seguimento a este projeto, mais pesquisas sobre os erros citados no capítulo 4, devem ser realizadas a fim de achar uma correção para os mesmos.

Uma outra sugestão é a de utilizar computadores de baixo custo para substituir o notebook utilizado como servidor Web, já que para esse serviço em uma residência não necessita de um poder de processamento muito elevado, porém atentando-se ao item 4.4.

E a última sugestão é a de integrar mais funções na automação já realizada, tentando abranger mais áreas da residência que podem ser automatizadas, como por exemplo:

- O controle da temperatura do ar condicionado;
- Sistema de detecção de presença;
- Sistema de alarme para vazamento de gás inflamável, como o gás butano;
- Controlar a luminosidade das lâmpadas tanto através do interruptor digital quanto pelo smartphone.

Referências

- Almeida, R. G. (2013). *A vida da fifi*. Brasília: Las Vegas.
- ANDROID. (2013). *Android Developer*. Acesso em 02 de Maio de 2014, disponível em Near Field Communication: <http://developer.android.com/guide/topics/connectivity/nfc/index.html>
- Angel, P. M. (1993). *INTRODUCCION A LA DOMOTICA*.
- Arduino.cc. (s.d.). Acesso em 21 de Abril de 2014, disponível em <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- ARGONAVIS. (2004). *Instituição de ensino e treinamento. Apresenta informações sobre diversas linguagens de programação*. Acesso em 02 de Maio de 2014, disponível em www.argonavis.com.br/cursos/web/w100/cws1_1.pdf
- AURESIDE. (2014). Acesso em 10 de Abril de 2014, disponível em Associação Brasileira de Automação Residencial: <http://www.aureside.org.br/temastec/default.asp?file=concbasicos.asp>
- Banzi, M. (2010). *Primeiros passos com o arduino* (Primeira ed.). São Paulo, SP, Brasil: Novatec Editora Ltda.
- Bolzani, C. A. (2004). *Residências inteligentes* (1ª ed.). (J. r. Marinho, Ed.) São Paulo: Editora Livraria da Física.
- Bortoluzzi, M. (10 de Janeiro de 2013). *SRA Engenharia Blog*. Acesso em 04 de 04 de 2014, disponível em http://sraengenharia.blogspot.com.br/2013/01/historico-da-automacao-residencial_10.html
- BROADCOM. (2012). *Near Field Communications (NFC) Controller Family*. Acesso em 3 de maio de 2014, disponível em BROADCOM: <http://www.broadcom.com/products/NFC/NFC-Solutions/BCM2079x-Family>
- Dall'Oglio, P. (2007). *Programando com orientação a objeto*. (R. Prates, Ed.) São Paulo: Novatec Editora Ltda.
- Hoag, M. (2002). *SERVIDORES WEB usando o APACHE*. (M. A. Denega, Ed.) São Paulo: Editora Berkeley.
- McRoberts, M. (2011). *Arduino Básico*. São Paulo, SP: Novatec.
- Monk, S. (2013). *Programação com Arduino >>começando com sketches*. Porto Alegre, RS: BOOKMAN EDITORA LTDA.
- Netcraft. (2014). Acesso em 27 de abril de 2014, disponível em April 2014 Web Server Survey: <http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html#more-14567>
- Pazos, F. (2002). *Automação de Sistemas & Robótica*. (R. Reinprecht, Ed.) Rio de Janeiro: Axcel Books do Brasil.

Pi, R. (14 de janeiro de 2014). Fonte: raspberrypi.org: <http://www.raspberrypi.org/>

Ramalho, J. A. (1996). *Iniciando em HTML*. (M. M. Filho, Ed.) São Paulo: Makron Books.

Rostli, J. (2013). *New Broadcom NFC chip, the consequences*. Acesso em 3 de Maio de 2014, disponível em NFC-Phones: <http://www.nfc-phones.org/broadcom-nfc-chip/>

Santos, D. M. (2013). *Relé*. Acesso em 3 de maio de 2014, disponível em InfoEscola: <http://www.infoescola.com/electronica/rele/>

TAKIUCHI, M., MELO, É., & TONIDANDEL, F. (2004). *DOMÓTICA INTELIGENTE: AUTOMAÇÃO BASEADA EM COMPORTAMENTO*. Centro Universitário da FEI – UniFE, São Bernardo do Campo - SP.

Thomazini, D., & Albuquerque, P. U. (2005). *Sensores Industriais* (1ª ed.). (R. A. Santos, Ed.) São Paulo: Érica.

WENSHING. (2013). *Data_Sheets*. Acesso em 3 de maio de 2014, disponível em WENSHING: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/General/TWS-BS-3_433.92MHz_ASK_RF_Transmitter_Module_Data_Sheet.pdf

APÊNDICE A – Página inicial do servidor Web

```

<html>

<head>

<meta http-equiv="Content-Language" content="pt-br">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">

<title>Automação com Arduino</title>

</head>

<body>

<div align="left">

    <table border="80" width="50%" cellpadding="20" >

        <tr></tr><tr><td>

<!-----POST DO PORTÃO----->

<form method="POST" action="">

<div align="center">

<A HREF="portao.php"><IMG SRC="portao.gif" BORDER="4"></A>

<p> </p>

</form>

        </td><td>

<!-----POST DO SALA----->

<form method="POST" action="">

<div align="center">

<A HREF="sala.php"><IMG SRC="salamin.gif" BORDER="4"></A>

<p>

</p>

</form>

        </td><td>

```

```
<!--POST DO BOTAO LIGAR AMBIENTE2-->
```

```
<form method="POST" action="">
```

```
<div align="center">
```

```
<p>
```

```
<input type="hidden" value="AMBIENTE2" name="estado" >
```

```
<input type="submit" value="AMBIENTE 2" name="AMBIENTE2" style="width:290;height:200" >
```

```
</p>
```

```
</form>
```

```
</td><td>
```

```
<!--POST DO BOTAO LIGAR AMBIENTE3-->
```

```
<form method="POST" action="">
```

```
<div align="center">
```

```
<p>
```

```
<input type="hidden" value="AMBIENTE3" name="estado" >
```

```
<input type="submit" value="AMBIENTE 3" name="AMBIENTE3" style="width:290;height:200" >
```

```
</p>
```

```
</form>
```

```
</td></tr><tr><td>
```

```
<!--POST DO BOTAO ACIONAR-->
```

```
<form method="POST" action="">
```

```
<div align="center">
```

```
<input type="hidden" value="fecha" name="estado" >
```

```
<input type="submit" value="Aciona" name="fecha" style="width:290;height:200" >
```

```
</form>
```

```
<!--POST DO BOTAO DESATIVAR-->
```

```
<form method="POST" action="">
```

```

<div align="center">

<p>

    <input type="hidden" value="desativa" name="estado" >

    <input type="submit" value="Desativa" name="desativa" style="width:290;height:200">

</p>

</form>

</table>

</div>

</body>

</html>

```

```

<?php

exec("mode com3: BAUD=9600 PARTITY=N data=8 stop=1 xon=off");

$port = fopen('COM1', 'w+');

if ($_POST['estado']=="abre")

{

    fwrite($port, "r");

}

if ($_POST['estado']=="lampada1")

{

    fwrite($port, "b");

}

if ($_POST['estado']=="lampada2")

{

    fwrite($port, "c");

}

```

```
if ($_POST['estado']=="lampada3")
{
    fwrite($port, "d");
}
if ($_POST['estado']=="fecha")
{
    fwrite($port, "a");
}
if ($_POST['estado']=="desativa")
{
    fwrite($port, "z");
}
fclose($port);
?>
```

APÊNDICE B – Código do controlador central

```
#include <IRremote.h>
#include <VirtualWire.h>
int portao = 12;
int iluminacao = 8;
char caracter;
void setup()
{
    Serial.begin(9600);
    pinMode(portao, OUTPUT);
    pinMode(iluminacao, OUTPUT);
    vw_set_ptt_inverted(true);
    vw_setup(2000);
    vw_rx_start();
}
void loop()
{
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;
    if (vw_get_message(buf, &buflen)) // Non-blocking
    {
        int i;
        for (i = 0; i < buflen; i++)
        {
            Serial.print(buf[i]);
            if(buf[i]==102)
            {
                digitalWrite(iluminacao, HIGH);
            }
            if(buf[i]==104)
            {
                digitalWrite(iluminacao, LOW);
            }
        }
    }
    caracter = Serial.read();
    if(caracter == 'r')
    {
        digitalWrite(portao, HIGH);
        delay(1000);
        digitalWrite(portao, LOW);
    }
}
```

```
    }  
    //=====  
    if(caracter == 'h')    // LIGA LUZ  
    {  
        digitalWrite(iluminacao, HIGH);  
    }  
    //=====  
    if(caracter == 'f')    // DESLIGA LUZ  
    {  
        digitalWrite(iluminacao, LOW);  
    }  
}
```

APÊNDICE C – Código do sistema autônomo de automação

```
#include <Wire.h>
#include <RTCLib.h>
#include <LiquidCrystal.h>

RTC_DS1307 RTC;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int piscina = 13;
int ponteH6 = 6;
int ponteH7 = 7;
int irrigando = 10;
int sensorUmidade = 0;
int umidade_val;
int saida_valor = 0;
int ct = 0;
void setup()
{
    Serial.begin(9600);
    Wire.begin();
    RTC.begin();
    pinMode(piscina, OUTPUT);
    pinMode(ponteH6, OUTPUT);
    pinMode(ponteH7, OUTPUT);
    pinMode(irrigando, OUTPUT);
    digitalWrite(piscina, LOW);
    lcd.begin(16, 2);
    if (!RTC.isrunning())
    {
        RTC.adjust(DateTime(__DATE__, __TIME__));
    }
}
void loop()
{
    DateTime agora = RTC.now();
    String relógio_data = "";
    String relógio_hora = "";
    int dia = agora.day();
    int mes = agora.month();
    int ano = agora.year();

    relógio_data += ArrumaZero(dia);
    relógio_data += "/";
    relógio_data += ArrumaZero(mes);
```

```

relogio_data += "/";
relogio_data += ano;

int hora = agora.hour();
int minuto = agora.minute();
int segundo = agora.second();

if(hora >= 20 && hora <= 21)
{
    digitalWrite(piscina,HIGH);
}
else
{
    digitalWrite(piscina,LOW);
}

relogio_hora += ArrumaZero(hora);
relogio_hora += ".";
relogio_hora += ArrumaZero(minuto);
relogio_hora += ".";
relogio_hora += ArrumaZero(segundo);

lcd.setCursor(0, 0);
lcd.print(relogio_data);
lcd.setCursor(0, 1);
lcd.print(relogio_hora);

//=====
umidade_val = analogRead(sensorUmidade);
saida_valor = map(umidade_val, 0, 1023, 100, 0);
if(saida_valor < 30 && ct == 0)
{
    digitalWrite(irrigando, HIGH);
    digitalWrite(ponteH6, HIGH);
    digitalWrite(ponteH7, LOW);
    delay(500);
    digitalWrite(ponteH6, LOW);
    digitalWrite(ponteH7, LOW);
    ct=1;
}
if(saida_valor > 30 && ct==1)
{
    digitalWrite(irrigando, LOW);
    digitalWrite(ponteH6, LOW);

```



```

        digitalWrite(ponteH7, HIGH);
        delay(500);
        digitalWrite(ponteH6, LOW);
        digitalWrite(ponteH7, LOW);
        ct=0;
    }
    if(saida_valor < 10)
    {
        lcd.setCursor(14, 1);
        lcd.print(" ");
        lcd.setCursor(15, 1);
        lcd.print(saida_valor);
    }
    else{
        lcd.setCursor(14, 1);
        lcd.print(saida_valor);
    }
    //=====
    delay(200);
}

String ArrumaZero (int i)
{
    String ret;
    if (i < 10)
    {
        ret += "0";
        ret += i;
    }
    return ret;
}

```

APÊNDICE D – Código do leitor de NFC

```
#include <PN532.h>
#define SCK 13
#define MOSI 11
#define SS 10
#define MISO 12

PN532 nfc(SCK, MISO, MOSI, SS);
int porta8 = 8;
int negado = 9;

void setup(void) {
  pinMode(porta8, OUTPUT);
  pinMode(negado, OUTPUT);
  Serial.begin(9600);
  nfc.begin();

  uint32_t versiondata = nfc.getFirmwareVersion();
  if (!versiondata) {
    Serial.print("Não foi encontrado um placa PN53x válida");
    while (1);
  }
  // mostra as configurações do dispositivo conectado
  Serial.print("Chip encontrado PN5"); Serial.println((versiondata>>24) & 0xFF, HEX);
  Serial.print("Versão do Firmware. "); Serial.print((versiondata>>16) & 0xFF, DEC);
  Serial.print('.'); Serial.println((versiondata>>8) & 0xFF, DEC);

  // configura o adaptador para ler cartões e tags de RFID
  nfc.SAMConfig();
}

void loop(void) {
  uint32_t id;
  id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);

  if(id == 2021683202 || id == 11184810 )
  {
    Serial.print("ID encontrada: "); Serial.println(id);
    digitalWrite(porta8, HIGH);
  }
}
```

```
    delay(1000);  
    digitalWrite(porta8, LOW);  
  }  
  if(id != 2021683202 && id != 11184810 && id != 0)  
  {  
    digitalWrite(negado, HIGH);  
    delay(1000);  
    digitalWrite(negado, LOW);  
  }  
  delay(500);  
}
```

APÊNDICE E – Código do interruptor digital

```
#include <VirtualWire.h>
#include <UTFT.h>
#include <UTouch.h>
#include <UTFT_Buttons.h>

// Declare which fonts we will be using
extern uint8_t BigFont[];

UTFT      myGLCD(TFT01_32,38,39,40,41);
UTouch    myTouch(6,5,4,3,2);
UTFT_Buttons myButtons(&myGLCD, &myTouch);
int ledPin13 = 13;
int ledPin10 = 10;
int ledPin11 = 11;
int ligarradio = 8;

void setup()
{
  Serial.begin(9600);
  Serial.println("setup");
  vw_set_ptt_inverted(true); // Required for DR3100
  vw_setup(2000);           // Bits per sec

  pinMode(ledPin13, OUTPUT);
  pinMode(ligarradio, OUTPUT);
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setFont(BigFont);

  myTouch.InitTouch();
  myTouch.setPrecision(PREC_MEDIUM);

  myButtons.setTextFont(BigFont);
}

void loop()
{
  int but1, but2, botao_apertado;
```

```

but1 = myButtons.addButton( 10, 20, 300, 70, "LIGAR");
but2 = myButtons.addButton( 10, 100, 300, 70, "DESLIGAR");
myButtons.drawButtons();
myGLCD.print("VOCE APERTO:", 10, 200);
myGLCD.setColor(VGA_BLACK);
myGLCD.setBackColor(VGA_WHITE);
myGLCD.print("      ", 10, 220);

while(1)
{
    if (myTouch.dataAvailable() == true)
    {
        botao_apertado = myButtons.checkButtons();
        if (botao_apertado==but1)
        {
            myGLCD.print("LIGAR      ", 10, 220);
            const char *msg = "f";
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx();
        }
        if (botao_apertado==but2)
        {
            myGLCD.print("DESLIGAR      ", 10, 220);
            const char *msg = "h";
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx();
        }
    }
}
}

```